

UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA DE TELECOMUNICACIÓN



PROYECTO FIN DE CARRERA

***ESQUEMAS DE CODIFICACIÓN AVANZADOS PARA
COMUNICACIONES ÓPTICAS***

AUTOR: ARÁNZAZU FERNÁNDEZ ÁLVAREZ

TUTOR: PABLO MARTÍNEZ OLMOS

SEPTIEMBRE DE 2014

TÍTULO: *Esquemas de codificación avanzados para comunicaciones ópticas*

AUTOR: *Aránzazu Fernández Álvarez*

TUTOR: *Pablo Martínez Olmos*

La defensa del presente Proyecto Fin de Carrera se realizó el día 10 de Septiembre de 2014; siendo calificada por el siguiente tribunal:

PRESIDENTE: *Matilde Sánchez Fernández*

SECRETARIO: *Javier Céspedes Martín*

VOCAL: *Pablo Acedo Gallardo*

Habiendo obtenido la siguiente calificación:

CALIFICACIÓN:

Presidente

Secretario

Vocal

Agradecimientos

En primer lugar a mi tutor, Pablo Martínez Olmos, por su inestimable ayuda, su interés constante y su trabajo, siempre entusiasta.

A mi familia, amigos y compañeros de estudios, sin cuyo apoyo continuo el trayecto habría sido mucho más complejo.

A Yolanda, por conseguir que recuerde con cariño las infinitas horas de trabajo. Y a Lucía, por el resto del tiempo.

*"Alice: How long is forever?
White Rabbit: Sometimes, just one second."*

-Lewis Carroll

Resumen

La demanda de mayores velocidades de transporte motiva el uso de novedosas técnicas avanzadas de modulación, detección, procesamiento de señal y codificación en los sistemas de comunicaciones ópticas de alta velocidad. En este proyecto, se revisa el uso de las técnicas correctoras de errores en los sistemas de comunicaciones de fibra óptica de ultra larga distancia y se muestra el potencial de utilizar técnicas de codificación avanzadas.

Los códigos basados en grafos, como los códigos de chequeo de paridad con baja densidad (*Low Density Parity Check*, LDPC), se han estandarizado en muchas aplicaciones. Entre los diferentes tipos de códigos LDPC, los códigos LDPC con acoplamiento espacial (*Spatially-Coupled* LDPC, SC-LDPC) son especialmente apropiados para los sistemas de comunicaciones ópticas. Se utiliza una representación basada en *protographs* para describir la construcción de los códigos SC-LDPC terminados, y se propone un decodificador enventanado que está basado en el algoritmo *Belief Propagation* (BP) y aprovecha la estructura de los códigos con acoplamiento. El esquema de decodificación enventanado proporciona una latencia y complejidad computacional más reducidos. Finalmente, se presentan los resultados obtenidos simulando la decodificación en ventanas sobre un canal que añade ruido blanco, gaussiano, comparándolos con esquemas FEC de primera, segunda y tercera generación.

Palabras clave: código de chequeo con baja densidad, acoplamiento espacial, códigos convolucionales, *belief propagation*, decodificación iterativa, decodificación enventanada.

Abstract

Demands for higher transport speed inspire the applications of novel advanced techniques in modulation, detection, signal processing and coding in high speed optical communication systems. In this Project, application of forward error correction techniques to long-haul fiber-optic communication systems is reviewed and the potential for applying advanced coding techniques is illustrated.

Codes on graphs, such as *Low Density Parity Check* (LDPC) codes, are becoming standard in many applications. Among different kinds of LDPC codes, *Spatially-Coupled* LDPC (SC-LDPC) codes are especially suitable for optical communication systems. We use a protograph representation to describe the construction of terminated SC-LDPC codes, and we propose a sliding *windowed decoder* that is based on the Belief-Propagation (BP) algorithm and exploits the structure of coupled codes. The windowed decoding scheme provides reduced latency and computational complexity. Finally, we give results obtained by simulating windowed decoding over an additive Gaussian noise channel, comparing them with first, second and third-generation FEC schemes.

Keywords: Low density parity check codes, spatial coupling, convolutional codes, belief propagation, iterative decoding, windowed decoding,

Índice general

1 INTRODUCCIÓN	1
1.1 Introducción a la codificación de canal	3
1.1.1 Sistemas de transmisión digital	3
1.1.2 Codificación de canal. Tipos de códigos	4
1.1.2.1 Prestaciones de las técnicas de protección contra errores	5
1.1.2.2 Tipos de códigos de protección contra errores	5
1.1.2.3 Códigos bloque	6
1.1.3 Decodificación de mínima probabilidad de error	8
1.1.4 Límites de la codificación de canal	10
1.2 Códigos LDPC	10
1.2.1 Códigos bloque	11
1.2.2 Códigos convolucionales	12
1.3 Metodología y objetivos del proyecto	12
1.4 Estructura del documento	13
 2 ESTADO DEL ARTE	 15
2.1 Red óptica de transporte	16
2.1.1 Beneficios del uso de técnicas FEC en sistemas de comunicaciones ópticas.	17
2.1.2 Funciones FEC en equipos terminales de comunicaciones ópticas	18
2.1.2.1 Recomendación UIT-T G.975	18
2.1.2.2 Recomendación UIT-T G.975.1	19
2.2 Evolución de las técnicas FEC	20
2.2.1 Generaciones de funciones FEC	20
2.2.1.1 Inicio de técnicas FEC en comunicaciones ópticas	21
2.2.1.2 Primera generación FEC	21
2.2.1.3 Segunda generación FEC	21
2.2.1.4 Tercera generación FEC	23
2.2.2 Códigos LDPC para tercera generación FEC	25

3 CODIFICACIÓN LDPC	29
3.1 Códigos bloque LDPC	31
3.1.1 Representación gráfica de la matriz de chequeo de paridad	31
3.1.2 Algoritmo de decodificación Belief Propagation	32
3.1.2.1. Paso de mensajes	33
3.1.2.2 Notación	33
3.1.2.3 Algoritmo de cociente de verosimilitudes	34
3.2 Códigos SC-LDPC	39
3.2.1 Estructura de los códigos SC-LDPC	39
3.2.2 SC-LDPC basado en protographs	40
3.2.3 Construcción de la matriz de chequeo de paridad	42
3.2.4 Decodificación en ventanas	45
 4 ANÁLISIS DE PRESTACIONES DE CÓDIGOS SC-LDPC	 49
4.1 Entorno de simulación	50
4.1.1 Entorno de simulación Matlab/MEX	50
4.1.2 Modelado del sistema de comunicaciones: simplificaciones	52
4.1.3 Arquitectura del emulador	52
4.1.4 Evaluación de las prestaciones	55
4.2 Análisis de códigos SC-LDPC con decodificación en ventanas	56
4.2.1 Propagación dura SC-LDPC(3,24)	58
4.2.2 Propagación blanda SC-LDPC(3,24)	60
4.2.2.1 Efecto de decodificación desde los extremos hacia el centro	60
4.2.2.2 Número de iteraciones	62
4.2.2.3 Tamaño de ventana	63
4.2.3 Códigos SC-LDPC con diferentes tasas. Comparativa con QC-LDPC	65
 5 CONCLUSIONES Y LÍNEAS FUTURAS	 71
5.1 Conclusiones	72
5.2 Líneas de trabajo futuro	74
5.2.1 Simulación software	74
5.2.2 Simulación hardware	75
 APÉNDICES	 77
A. Glosario de acrónimos y nomenclatura	79
A.1 Acrónimos	79
A.2 Nomenclatura	80

B. Glosario de reglas matemáticas	83
<i>B.1 Regla de Bayes</i>	83
<i>B.2 Regla de la tangente</i>	83
C. Tiempos de decodificación	85
<i>C.1 SC-LDPC(3,24), L=48</i>	85
<i>C.2 SC-LDPC(4,32), L=47</i>	85
<i>C.3 SC-LDPC(3,15), L=48</i>	86
<i>C.4 SC-LDPC(4,20), L=47</i>	87
 BIBLIOGRAFÍA	 89

Lista de Figuras

1.1	Diagrama de bloques de un sistema de transmisión digital	3
1.2	Diagrama de bloques de codificación/decodificación de canal	4
1.3	Prestaciones de códigos LDPC-BC, para transmisión BPSK [2]	11
2.1	Diagrama de bloques de un sistema óptico. G.975	18
2.2	Diagrama de bloques de un sistema óptico DWDM. G.975.1	19
2.3	Progreso FEC en las últimas décadas [8]	20
2.4	Comparativa BER en función de E_b/N_0 . Transmisión BPSK, canal AWGN [19]	26
2.5	Estructura de la matriz de chequeo de un código SC-LDPC	27
3.1	Grafo de Tanner construido a partir de H	31
3.2	Diagrama de bloques de codificación/decodificación LDPC	32
3.3	Intercambio de mensajes, en algoritmo de paso de mensajes	33
3.4	Protograph de un LDPC-BC(3,6), $n_v = 2$, $n_c = 1$, $R = \frac{1}{2}$	41
3.5	Cadena de protographs no acoplados	41
3.6	Cadena de protographs acoplados	42
3.7	Estructura de un código SC-LDPC terminado	42
3.8	Matriz H de un SC-LDPC (3,6), M=1000, L=20	44
3.9	Grado de los nodos de chequeo en el código de la figura 3.8	45
3.10	Ventana de decodificación en quinta iteración, W=4, ms=2	46
3.11	Ventana de decodificación en sexta iteración, W=4, ms=2	47
4.1	Diagrama de bloques del sistema de comunicaciones modelado con Matlab/MEX	53
4.2	Comparativa propagación soft/hard, para SC-LDPC(3,24), L=48, 20 iteraciones	58
4.3	Comparativa propagación hard, para SC-LDPC(3,24), L=48	59
4.4	LLR tras un número variable de iteraciones, con decodificación bloque	60
4.5	LLR tras un número variable de iteraciones, con decodificación en ventanas	61
4.6	Comparativa propagación soft, variación it, para SC-LDPC(3,24), L=48.	62
4.7	Comparativa propagación soft, variación W, para SC-LDPC(3,24), L=48, 20 iteraciones	63
4.8	Comparativa propagación soft, variación W, para SC-LDPC(3,24), L=48, 40 iteraciones	64
4.9	Comparativa SC-LDPC con diferentes grados	65
4.10	Comparativa SC-LDPC (4,32)	66
4.11	Comparativa SC-LDPC con QC-LDPC[19] y [15], y códigos de primera y segunda generación	68

Lista de Tablas

2.1	Prestaciones de los esquemas FEC de segunda generación G.975.1 concatenados	22
2.2	Prestaciones de los esquemas FEC de segunda generación G.975.1 no concatenados	23
3.1	Características de códigos LDPC en función de la distribución de los grados	30
4.1	Tasas de los códigos SC-LDPC propuestos	56
4.2	Longitudes de las palabras código en función de M y L	56
4.3	Longitudes de las palabras a decodificar en función de M y W	57
C.1	Tiempos de decodificación de palabra código SC-LDPC(3,24) L=48, 20 y 40 iteraciones	86
C.2	Tiempos de decodificación de palabra código SC-LDPC(4,32) L=47, 20 y 40 iteraciones	86
C.3	Tiempos de decodificación de palabra código SC-LDPC(3,15) L=48, 20 y 40 iteraciones	87
C.4	Tiempos de decodificación de palabra código SC-LDPC(4,20) L=47, 20 y 40 iteraciones	88

1

Introducción

Contenidos

1.1 Introducción a la codificación de canal.....	3
1.1.1 Sistemas de transmisión digital	
1.1.2 Codificación de canal. Tipos de códigos	
1.1.3 Decodificación de mínima probabilidad de error	
1.1.4 Límites de la codificación de canal	
1.2 Códigos LDPC.....	10
1.2.1 Códigos bloque	
1.2.2 Códigos convolucionales	
1.3 Metodología y objetivos del proyecto.....	12
1.4 Estructura del documento.....	13

El mercado de las Telecomunicaciones se encuentra en constante evolución y crecimiento. A lo largo de los últimos años se han desplegado nuevos servicios, que junto con los ya existentes han desencadenado un crecimiento exponencial del volumen de tráfico multimedia, de voz y datos. En un contexto en el que la demanda de ancho de banda crece de modo constante es preciso desarrollar nuevas tecnologías de red para conseguir mayores tasas de transmisión: ya se ha estandarizado 100 Gigabit Ethernet, y las previsiones de las futuras demandas de ancho de banda hacen que se esté explorando el desarrollo de un nuevo estándar 400 Gigabit Ethernet. La gran difusión de los servicios Ethernet, Internet y de elevada velocidad ha originado que las redes de transporte evolucionen hacia tecnologías ópticas y de paquetes. La Red Óptica de Transporte (*Optical Transport Network*, OTN), que permite combinar diferentes servicios y redes, surge como tecnología de transporte de nueva generación para redes de fibra óptica.

Se considera que el uso de un código de detección y corrección de errores en recepción es una de las claves que habilitará la implementación de las redes ópticas de próxima generación, ya que permite incrementar la capacidad de transmisión y reducir costes. Hoy en día en la práctica se utiliza una amplia gama de códigos introducidos con el objetivo de optimizar la transmisión, pero la falta de homogeneidad no favorece la interoperabilidad entre dispositivos y redes. El éxito de la migración hacia una infraestructura 100G y de tasas superiores reside en gran medida en la estandarización de sus soluciones, lo que ha motivado que en la actualidad se encuentre en auge el estudio de nuevos esquemas de codificación avanzados.

Entre la gran variedad de códigos existentes se ha de seleccionar un código con el que se obtenga una ganancia de codificación más adecuada para las nuevas tasas de transmisión y que se adapte a las características, necesidades y objetivos de los sistemas de comunicaciones ópticas. Dentro de la extensa gama cabe destacar a los códigos LDPC (*Low Density Parity Check codes*) como firmes candidatos para ser introducidos en los próximos estándares de las redes ópticas de transporte.

1.1 Introducción a la codificación de canal

1.1.1 Sistemas de transmisión digital

En la *figura 1.1* se representa el modelo de un sistema de comunicaciones digital:

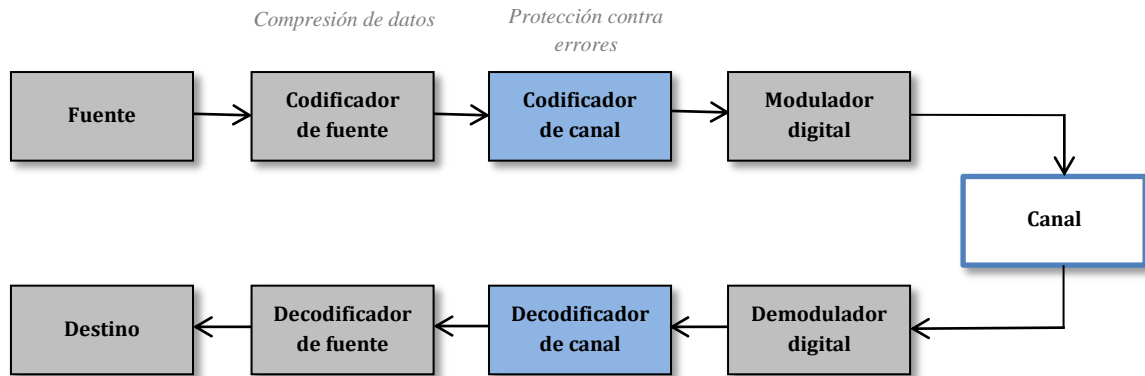


Figura 1.1: Diagrama de bloques de un sistema de transmisión digital

El sistema de transmisión está compuesto por los siguientes bloques funcionales:

Fuente de información digital: genera los datos a transmitir, representados en formato digital.

Codificador de fuente: comprime los datos ofrecidos por la fuente de información eliminando parte de la redundancia ofrecida por ésta. Típicamente la redundancia introducida por la fuente no es estructurada, por lo que no protege de modo uniforme la información ni proporciona protección contra errores.

Codificador de canal: añade redundancia estructurada con la finalidad de realizar en recepción detección o corrección de errores introducidos durante el proceso de transmisión. En esta etapa se agrega redundancia de modo controlado, proporcionando capacidad de control de errores.

Se distinguen dos conjuntos de estrategias de control de errores en función de cómo se realice el tratamiento de errores en recepción:

- **Técnicas ARQ, Solicitud de Repetición Automática** (*Automatic Repeat reQuest*): utilizan la redundancia para realizar exclusivamente detección de errores. Si el receptor detecta un error solicita el reenvío del mensaje.

1. Introducción

- **Técnicas FEC, Corrección de Errores hacia Adelante** (*Forward Error Correction*): la redundancia es utilizada para detectar y corregir errores en recepción. Los esquemas de codificación basados en códigos LDPC estudiados en este proyecto pertenecen a este grupo.

Modulador digital: convierte el flujo de símbolos de salida del codificador de canal en señales con forma de onda adecuada para ser transmitidas a través del canal.

Canal: es el medio utilizado para transmitir la información.

Demodulador digital: recibe la señal del canal y la convierte en una secuencia de símbolos. Esta etapa suele incluir diversas funciones: filtrado, demodulación, sincronización de portadora, sincronización de trama...

Decodificador de canal: utiliza la redundancia introducida por el codificador de canal para detectar o corregir errores introducidos por el canal.

Decodificador de fuente: proporciona una representación no comprimida de los datos.

1.1.2 Codificación de canal. Tipos de códigos

En la *figura 1.2* se presenta el diagrama de bloques del proceso de codificación y decodificación. El codificador de canal acepta los bits del mensaje ofrecidos por la fuente de información y añade redundancia aplicando una transformación: asigna a cada una de las 2^k posibles palabras de entrada \mathbf{b} , de k símbolos de longitud pertenecientes a un alfabeto \mathcal{X} , una palabra código \mathbf{c} de n símbolos.

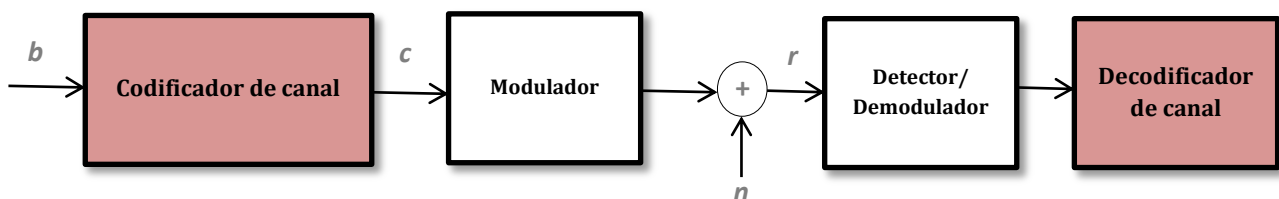


Figura 1.2: Diagrama de bloques de codificación/decodificación de canal

1. Introducción

En este proyecto se utilizan códigos binarios, de modo que el alfabeto $\mathcal{X} = \{0, 1\}$, y se emplea un decodificador con **entrada blanda**: al contrario que los sistemas con **decodificación dura**, en los que se decide sobre cada bit recibido antes de realizar el proceso de decodificación, el decodificador opera con la palabra recibida \mathbf{r} .

1.1.2.1 Prestaciones de las técnicas de protección contra errores

Existen diversas medidas para comparar las prestaciones de las técnicas de protección contra errores:

- **Ganancia de codificación:** es la diferencia en *decibelios* entre las relaciones E_b/N_0 – relaciones energía por bit a ruido – necesarias para obtener una determinada tasa de error de bit (*Bit Error Rate, BER*) con y sin el esquema de protección.
- **Aumento de ancho de banda:** la tasa de bits a la salida del codificador es más alta que la tasa de entrada. En un esquema de modulación fijo la adición de redundancia implica un aumento del ancho de banda requerido en transmisión.

A la hora de diseñar un esquema de codificación no sólo se tiene en cuenta la **ganancia de codificación**: se ha de alcanzar una solución de compromiso con el **aumento de ancho de banda** y la **complejidad** del esquema (especialmente en el proceso de decodificación).

1.1.2.2 Tipos de códigos de protección contra errores

Son muchos los tipos y familias de códigos diseñados e implementados a lo largo de la historia. Tradicionalmente se han clasificado en dos grandes grupos: códigos bloque y códigos convolucionales. La principal diferencia entre ambos es el uso de *memoria*:

- **Códigos bloque:** el codificador opera con mensajes sucesivos de k símbolos y añade una redundancia de $\mathbf{r}_{cod} = n - k$ símbolos. La palabra código de salida depende únicamente del mensaje de entrada, por lo que se dice que estos códigos *no tienen memoria*.

1. Introducción

- **Códigos convolucionales:** igual que el codificador bloque, el codificador convolucional acepta mensajes sucesivos de k símbolos. La diferencia fundamental es que la palabra código de salida de longitud n no depende únicamente del bloque de entrada de k símbolos, sino también de los m bloques anteriores. Se dice que el código convolucional tiene *memoria de orden m* .

En ambos casos la relación entre la longitud de las palabras de entrada y las palabras código se conoce como tasa de código:

$$tasa\ de\ código = R = \frac{k}{n} = \frac{n - r_{cod}}{n} = 1 - \frac{r_{cod}}{n} \quad (R < 1) \quad (1.1)$$

1.1.2.3 Códigos bloque

La operación de codificación en el caso de los códigos bloque se corresponde con una operación matricial. Un código bloque es **lineal** si las palabras código se obtienen a partir de la aplicación de una transformación lineal del mensaje de entrada.

La **distancia de Hamming d_h** es el número de elementos en los que difieren dos palabras código. Se conoce como **distancia mínima d_{min}** a la menor de todas ellas:

$$d_{min} = \min \left(d_h(c_i, c_j) \right) \quad i, j = 0, 1, \dots, 2^k - 1, i \neq j \quad (1.2)$$

Los códigos bloque lineales cumplen las siguientes propiedades:

- Cualquier combinación lineal de dos palabras código (suma aritmética módulo 2) es otra palabra código: si c_i y c_j son palabras código, $c_i \oplus c_j$ también lo es.
- El código incluye una palabra con todos los símbolos iguales a 0 ($c_i \oplus c_i = \bar{0}$).
- La d_{min} es igual al menor número de *unos* de las palabras código distintas de la palabra $\bar{0}$.
- Todas las palabras código tienen al menos otra palabra código a una distancia d_{min} .

1. Introducción

- **Matriz generadora G**

Siendo \mathbf{b} un bloque de k símbolos de mensaje,

$$\mathbf{b} = [b_0 \ b_1 \ \dots \ b_{k-1}] \quad (1.3)$$

la palabra código \mathbf{c} ,

$$\mathbf{c} = [c_0 \ c_1 \ \dots \ c_{n-1}] \quad (1.4)$$

se obtiene multiplicando el mensaje por una matriz generadora \mathbf{G} de dimensiones $[k] \times [n]$:

$$\mathbf{c} = \mathbf{bG} \quad (1.5)$$

Cada elemento del conjunto de palabras código $C = \{c_0, c_1, \dots, c_{2^k-1}\}$ es una combinación lineal de las filas de \mathbf{G} , que son linealmente independientes. Cada mensaje de entrada se mapea en una única palabra de este conjunto.

En el caso de un **código sistemático** los k primeros símbolos de las palabras código son iguales a los k símbolos que componen el mensaje y los $n - k$ restantes son la redundancia añadida. La palabra código es por tanto:

$$\mathbf{c} = [c_0 \ c_1 \ \dots \ c_{n-1}] = [b_0 \ b_1 \ \dots \ b_{k-1}, m_0 \ \dots \ m_{n-k-1}] \quad (1.6)$$

La matriz generadora de este tipo de códigos tiene la siguiente forma:

$$G = [P | I_k] \quad (1.7)$$

siendo \mathbf{P} una matriz de tamaño $[k] \times [n - k]$ e \mathbf{I}_k la matriz identidad $[k] \times [k]$.

1. Introducción

- **Matriz de comprobación de paridad H**

Se denomina matriz de comprobación de paridad H a la matriz generadora del complemento ortogonal del código, que es a su vez también un código bloque lineal:

$$GH^T = \bar{0} \quad (1.8)$$

Esta matriz tiene dimensiones $[n - k] \times [n]$. Su nombre deriva del hecho de que es utilizada para verificar si una palabra pertenece al código, ya que con cualquier palabra que pertenece a este conjunto se cumple la siguiente ecuación:

$$cH^T = bGH^T = \bar{0} \quad (1.9)$$

En el caso de un código sistemático:

$$H = [I_{n-k} | P^T] \quad (1.10)$$

donde I_{n-k} es la matriz identidad de dimensiones $[n - k] \times [n - k]$.

1.1.3 Decodificación de mínima probabilidad de error

El decodificador óptimo – el que proporciona una probabilidad de error mínima – se obtiene aplicando uno de los dos criterios *Máximo A Posteriori* (MAP) [1], que proporcionan prestaciones similares: el criterio MAP de bloque y el criterio MAP de bit.

Siendo \mathbf{c}^* la palabra transmitida perteneciente a un código $C = \{c_0, c_1, \dots, c_{2^k-1}\}$ y \mathbf{r} la palabra recibida de entrada del decodificador, con una longitud de n bits:

$$\mathbf{r} = [r_0 \ r_1 \ \dots \ r_{n-1}] \quad (1.11)$$

- **Decodificador MAP bloque**

El decodificador MAP bloque estima el mensaje $\hat{\mathbf{c}}$ que minimiza la probabilidad de error de bloque:

$$P(\hat{\mathbf{c}} \neq \mathbf{c}^*) \quad (1.12)$$

1. Introducción

La salida del decodificador es el mensaje $\hat{\mathbf{c}}$ que maximiza la *probabilidad a posteriori*:

$$\hat{\mathbf{c}}_{\text{MAP}} = \arg \max_{\mathbf{c} \in \mathcal{C}} p(\mathbf{c}/\mathbf{r}) \quad (1.13)$$

Si todas las palabras código son equiprobables el criterio MAP es equivalente al criterio de **Máxima Verosimilitud** (*Maximum Likelihood*, ML) equivalente a su vez a encontrar la palabra del código que se encuentra a menor distancia euclídea de la palabra recibida:

$$\hat{\mathbf{c}}_{\text{ML}} = \arg \max_{\mathbf{c} \in \mathcal{C}} p(\mathbf{r}/\mathbf{c}) \quad (1.14)$$

Este decodificador tiene complejidad $O(2^n)$, ya que se evalúa $p(\mathbf{c}/\mathbf{r})$ para el código completo.

- **Decodificador MAP de bit**

El decodificador MAP de bit, también con complejidad $O(2^n)$, estima el mensaje $\hat{\mathbf{c}}$ que minimiza la probabilidad de error de bit:

$$\frac{1}{n} \sum_{i=1}^n P(\hat{c}_i \neq c_i^*) \quad (1.15)$$

El bit i -ésimo de la palabra estimada se obtiene:

$$\hat{c}_{i\text{MAP}} = \arg \max_{c_i \in \{0,1\}} p(c_i|\mathbf{r}) \quad (1.16)$$

donde la probabilidad marginal *a posteriori* se calcula del siguiente modo:

$$p(c_i|\mathbf{r}) = \sum_{\mathbf{c} \setminus c_i} p(\mathbf{c}/\mathbf{r}) \quad (1.17)$$

El factor que limita la capacidad de corrección de errores en el decodificador óptimo es la complejidad de decodificación: crece exponencialmente con la longitud de la palabra código, lo que causa que sea inviable aplicar este criterio en códigos con palabras de longitudes elevadas. Esto justifica el uso de codificadores y decodificadores que pueden ser implementados con una complejidad moderada y que utilizan métodos computacionalmente más eficientes, a pesar de no ser los óptimos en términos de la probabilidad de error. En este trabajo se utiliza una instancia de un algoritmo de decodificación iterativo de paso de mensajes, el algoritmo *Belief Propagation* (BP), detallado en el capítulo tercero.

1.1.4 Límites de la codificación de canal

Shannon formula a finales de los años 40 los principios de la **Teoría de la Información**, expuestos en el artículo “*A Mathematical Theory of Communication*”. Dentro de la Teoría de la Información se enmarca el **Teorema de la codificación de canal**, en el que Shannon demuestra que si se transmite a una tasa inferior a la **Capacidad de canal C** existe un código con el que la probabilidad de error puede hacerse arbitrariamente pequeña.

La Teoría de la Información define una transmisión como fiable si “*para todo $\epsilon > 0$ existe una secuencia de códigos $[n, nR]$ y un valor n_o para el que sus probabilidades de error $P_e(máx, n) < \epsilon$ cuando $n > n_o$* ”. Es posible transmitir de modo fiable siempre que la tasa de transmisión sea menor que el límite **C**, que es una propiedad exclusiva del canal. El Teorema determina que existe un límite pero no cómo alcanzarlo.

1.2 Códigos LDPC

Los códigos de **Chequeo de Paridad de Baja Densidad** (*Low Density Parity Check codes*, LDPC) son originalmente propuestos por Gallager en 1962 y posteriormente olvidados debido a que su uso no era práctico en ese momento, ya que se requieren códigos con grandes longitudes. Fueron redescubiertos a principios de los años 90 al demostrarse que las prestaciones ofrecidas por estos códigos pueden acercarse al límite de Shannon con una baja complejidad si se utilizan algoritmos de decodificación iterativos. Desde entonces ha habido numerosos avances en el diseño de códigos LDPC y de algoritmos de decodificación.

Los códigos LDPC se describen a través de su matriz de chequeo de paridad **H**. La distribución de los elementos de esta matriz da nombre a los códigos: se caracteriza por ser muy dispersa, es decir, tiene un pequeño número de unos en comparación con la cantidad de ceros.

Hoy en día son muchos los sistemas de comunicaciones que incluyen códigos LDPC en sus estándares: LANS inalámbricas (IEEE 802.11n), WiMAX (IEEE 802.16e), Ethernet 10GBase-T (IEEE 802.3an), Broadcast de vídeo digital (DVB-S2)...

Pueden clasificarse en dos grandes grupos: **códigos bloque** (LDPC *Block Codes*, LDPC-BC) y **convolucionales** (LDPC *Convolutional Codes*, LDPC-CC).

1.2.1 Códigos bloque

A su vez los códigos bloque pueden subdividirse en otros dos conjuntos: códigos bloque **regulares** e **irregulares**. A continuación se describe para ambos casos la aparición de un fenómeno conocido como **error floor**, una disminución brusca de la pendiente de la curva BER, y el comportamiento en la región **waterfall**, la región de la curva BER anterior a la aparición de **error floor**.

En la *figura 1.3* [2] se representa la curva de probabilidad de error de bit en función de la relación E_b/N_0 para cada uno de ellos:

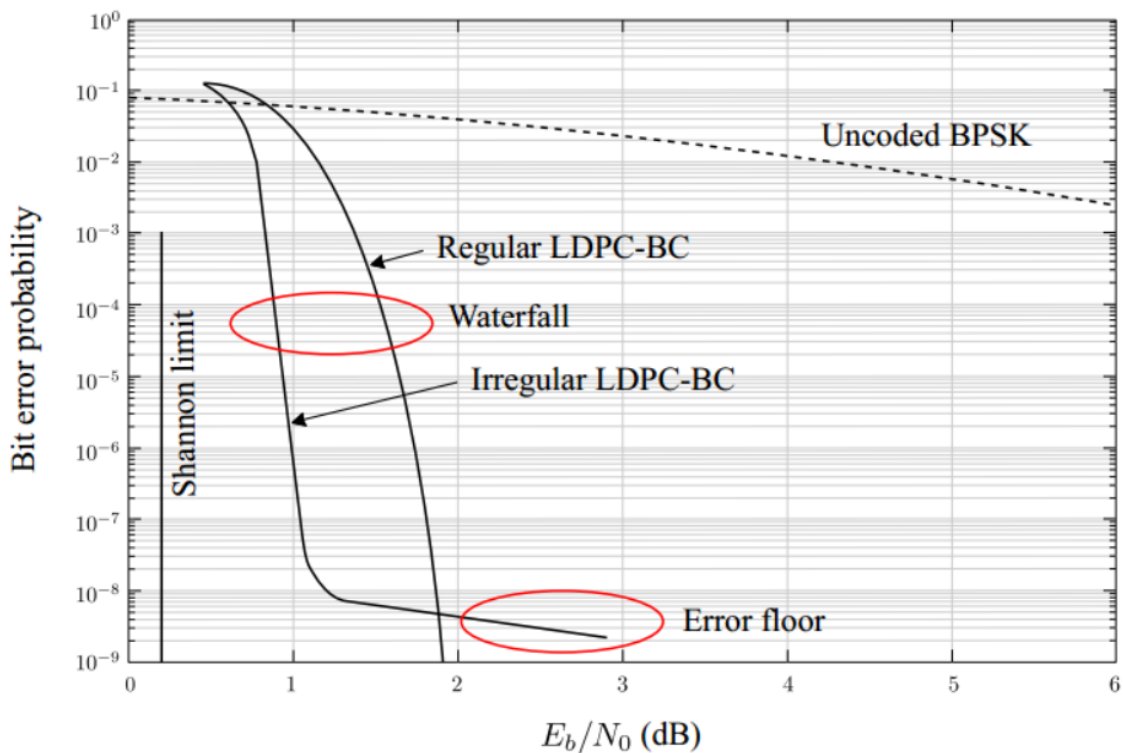


Figura 1.3: Prestaciones de códigos LDPC-BC, para transmisión BPSK [2]

- **Códigos bloque regulares**

Los códigos LDPC regulares son aquellos que tienen en la matriz **H** la misma cantidad de unos en cada una de las columnas, y lo mismo ocurre con el número de unos en cada fila (todas las columnas tienen el mismo **grado**, al igual que las filas). Esta estructura regular facilita el diseño de la arquitectura del codificador y decodificador y su implementación en un circuito integrado.

En la región **waterfall** la curva BER tiene una pendiente más suave que los códigos irregulares, no se aproximan tanto al límite de Shannon: se ha demostrado que las prestaciones ofrecidas por éstos se encuentran más lejos del límite al utilizar el algoritmo de decodificación iterativo BP

1. Introducción

[3]. Esta característica hace que no sean apropiados para sistemas de *broadcast* por satélite o transmisión móvil en el enlace ascendente, ya que estas aplicaciones están sujetas a limitaciones de potencia y deben conseguirse las mejores prestaciones posibles con relaciones señal a ruido moderadas.

- **Códigos bloque irregulares**

No todas las columnas y filas de la matriz de chequeo de paridad de los códigos irregulares tienen el mismo *grado*.

La BER de los códigos irregulares se aproxima al límite de Shannon en la región *waterfall* utilizando el algoritmo de decodificación BP [3] pero se manifiesta el fenómeno de *error floor*, lo que hace que por sí solos no sean idóneos para aplicaciones de comunicaciones ópticas que requieren BER muy bajas.

1.2.2 Códigos convolucionales

Los códigos LDPC-CC introducen memoria en el proceso de codificación. Este proyecto se centra en una nueva clase de códigos LDPC convolucionales: los códigos LDPC con **Acoplamiento Espacial** (*Spatially Coupled* LDPC, SC-LDPC). Se espera que estos códigos combinen los beneficios de los códigos LDPC-BC regulares e irregulares en las regiones de *waterfall* y *error floor*.

1.3 Metodología y objetivos del proyecto

En primer lugar se ha realizado una búsqueda bibliográfica para identificar el estado del arte de los códigos correctores de errores en los sistemas de comunicaciones ópticas con el objetivo de conocer sus aplicaciones, las prestaciones de los códigos ya estandarizados, los requisitos que han de cumplir los códigos de próxima generación y determinar las líneas de investigación y tendencias actuales.

Una vez conocidas las tendencias en el desarrollo de las técnicas FEC, se estudian en particular los códigos SC-LDPC con decodificación en ventanas y las ventajas que ofrecen. El objetivo final del proyecto es desarrollar una plataforma de simulación *software* que permite analizar las prestaciones de estos códigos emulando un sistema de comunicaciones simplificado, y verificar

que incluso con una construcción de la matriz de chequeo de paridad sencilla son competitivos con otros códigos propuestos hasta la fecha.

1.4 Estructura del documento

El resto del documento se estructura del siguiente modo:

En el segundo capítulo se presenta cómo se incorporan las técnicas FEC en los sistemas de comunicaciones ópticas y su importancia. A continuación se muestra la evolución que han tenido estas técnicas a lo largo de las últimas décadas y se plantean las necesidades de los sistemas de próxima generación, así como algunas de las soluciones propuestas basadas en códigos LDPC.

En el capítulo tercero se elabora un análisis de los códigos bloque LDPC y el algoritmo *Belief Propagation* utilizado en decodificación. Posteriormente se describe la estructura de los códigos SC-LDPC, el procedimiento de construcción de la matriz de chequeo y el proceso de decodificación en ventanas.

En el cuarto capítulo se expone el entorno *software* desarrollado para evaluar los códigos y se analizan las prestaciones de diversos códigos mediante los resultados obtenidos en simulación, generados teniendo en cuenta la máxima redundancia recomendada en el ámbito de las comunicaciones ópticas.

Finalmente, en el capítulo quinto se efectúa un análisis de los resultados obtenidos y se proponen posibles líneas de trabajo futuro.

2

Estado del Arte

Contenidos

2.1 Red Óptica de Transporte.....	16
2.1.1 Beneficios del uso de técnicas FEC en sistemas de comunicaciones ópticas.	
2.1.2 Funciones FEC en equipos terminales de comunicaciones ópticas	
2.1.2.1 Recomendación UIT-T G.975	
2.1.2.2 Recomendación UIT-T G.975.1	
2.2 Evolución de las técnicas FEC.....	20
2.2.1 Generaciones de funciones FEC	
2.2.1.1 Inicio de técnicas FEC en comunicaciones ópticas	
2.2.1.2 Primera generación FEC	
2.2.1.3 Segunda generación FEC	
2.2.1.4 Tercera generación FEC	
2.2.2 Códigos LDPC para tercera generación FEC	

La creciente demanda de ancho de banda y el incremento del volumen de tráfico a nivel mundial motivan el uso de tecnologías de transporte flexibles, que faciliten la interconexión e interoperabilidad entre múltiples redes utilizando una infraestructura común. Actualmente las tecnologías basadas en fibra óptica son las más utilizadas en las redes de transporte de alta velocidad.

En este capítulo se exponen en primer lugar la importancia y aplicaciones de las técnicas FEC en las redes de transporte óptico. A continuación se analiza la evolución de estas técnicas y las líneas de investigación actuales orientadas a las redes de próxima generación, basadas en códigos LDPC.

2.1 Red Óptica de Transporte

La Unión Internacional de Telecomunicaciones (UIT) ha desempeñado un papel muy activo en la normalización de las redes de transporte óptico. La UIT-T define **la Red Óptica de Transporte** (*Optical Transport Network*, OTN) mediante un conjunto de Recomendaciones, elaboradas en torno a la *Recomendación UIT-T G.709* [4]. La OTN - también conocida como “*Digital Wrapper*” o “*Contenedor Digital*”- es una estructura de transporte de elevada capacidad y largo alcance que permite combinar diferentes redes y servicios en una única infraestructura. Se trata de una tecnología compuesta por una serie de elementos de red conectados a través de enlaces de fibra óptica que ofrece funcionalidades de transporte, enrutado, multiplexación, gestión y supervisión de los canales ópticos.

Hasta el momento en las Recomendaciones de la UIT-T se incluyen tasas de transmisión de 2.5, 10, 40 y 100 Gb/s. Las necesidades actuales incentivan un inminente despliegue de la infraestructura 100G y la estandarización de tasas de transmisión superiores. Una de las soluciones menos costosas para incrementar la tasa de modo inmediato consiste en desarrollar los sistemas 100G utilizando la infraestructura ya existente para 10G. No obstante las prestaciones de los sistemas se ven fuertemente degradadas al incrementar la velocidad de transmisión: un incremento de la tasa de transmisión en un determinado canal se traduce en una reducción directa de la relación señal a ruido óptica (*Optical Signal to Noise Ratio*, OSNR), causando un crecimiento inasumible de la probabilidad de error.

2. Estado del arte

Es necesaria por tanto la utilización de diferentes herramientas avanzadas que faciliten conseguir unos niveles de calidad de transmisión aceptables con un coste asumible y la menor OSNR posible:

- Formatos de modulación óptimos.**
- Técnicas de detección y procesado de señal.**
- Técnicas de codificación y corrección de errores (FEC).**

2.1.1 Beneficios del uso de técnicas FEC en sistemas de comunicaciones ópticas

Las prestaciones de los sistemas de comunicaciones ópticas de larga distancia y alta velocidad se degradan considerablemente en términos de la probabilidad de error de bit al aumentarse la tasa de transmisión. Durante los últimos años se ha conseguido disminuir la OSNR necesaria para alcanzar una determinada probabilidad de error combinando esquemas de modulación cada vez más sofisticados y técnicas FEC más robustas: el uso de un código de corrección de errores en recepción es fundamental para incrementar la capacidad de transmisión y reducir costes, y es una de las claves que permitirá implementar las redes ópticas de próxima generación.

La incorporación de funciones FEC en los equipos terminales tiene implícito el aumento de la velocidad binaria de línea. A cambio se obtienen diversas ventajas:

- **Ganancia en el balance de potencia óptica total:** esta mejora facilita la disminución del coste total del sistema. Permite aumentar la distancia máxima de transmisión en el caso de los sistemas sin repetidores, y en el caso de los sistemas con amplificación aumentar las distancias totales entre los repetidores o relajar las especificaciones de los componentes. También es posible recortar la potencia de transmisión.
- **Decrece el valor mínimo de la tasa de error de bit (BER) de línea del sistema:** se mejora la calidad de la comunicación, ya que la información es protegida frente a posibles fallos de los componentes integrantes del sistema.

2.1.2 Funciones FEC en equipos terminales de comunicaciones ópticas

El uso de funciones FEC no es obligatorio en los equipos terminales ópticos de transmisión aunque resultan indispensables, especialmente cuando se utiliza Multiplexado por División en Longitud de onda (*Wavelength Division Multiplexing*, WDM) y amplificación óptica.

Los procedimientos de codificación y decodificación de información se realizan a nivel de los equipos terminales: en los sistemas ópticos los esquemas FEC se incorporan a través de un codificador FEC en el equipo terminal transmisor y un decodificador FEC en el equipo terminal receptor. El codificador acepta bits de información, añade redundancia a la señal y proporciona a su salida datos codificados a una tasa binaria más elevada que la tasa de entrada. En recepción el decodificador utiliza la redundancia enviada para realizar evaluación y corrección de errores, y finalmente la elimina para tratar de recuperar los datos de entrada del codificador.

Existen dos Recomendaciones de la UIT-T en las que se detalla el esquema de un sistema de transmisión óptico que incorpora funciones FEC: la **Recomendación UIT-T G.975** [5] para códigos FEC de primera generación y la **Recomendación UIT-T G.975.1** [6] en la que se describen los códigos FEC de segunda generación.

2.1.2.1 Recomendación UIT-T G.975

En la *figura 2.1* se muestra el sistema descrito en la Recomendación UIT-T G.975:

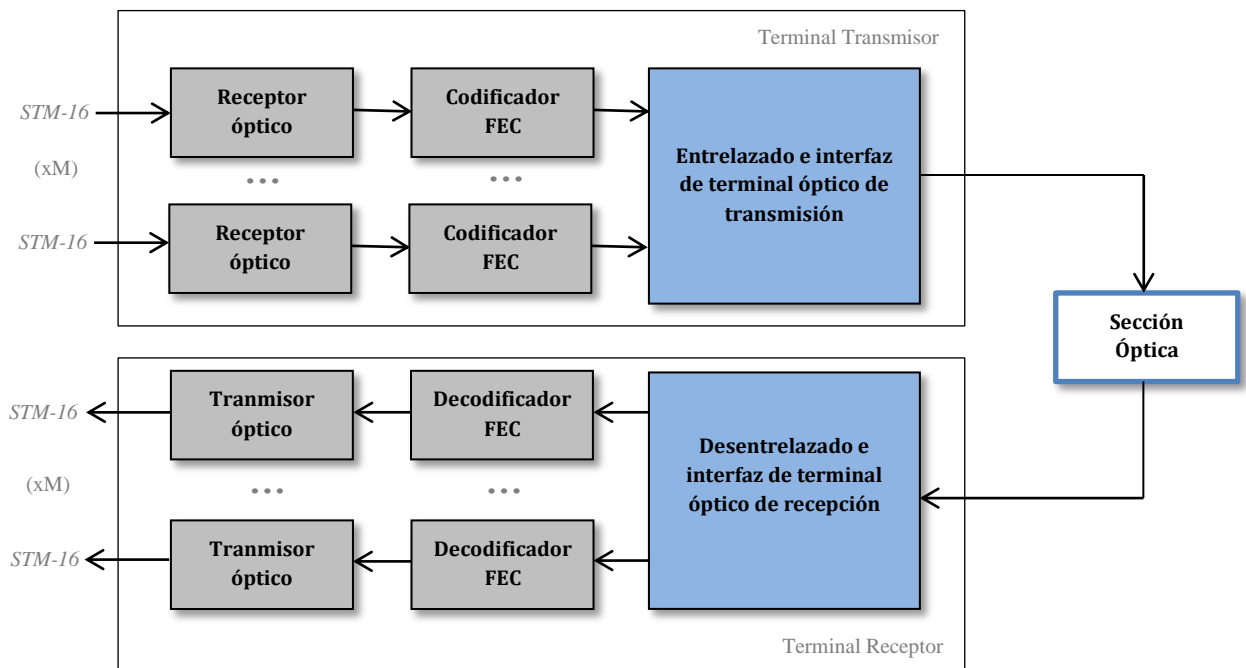


Figura 2.1: Diagrama de bloques de un sistema óptico. G.975

2. Estado del arte

Este sistema utiliza señales STM-16, con una velocidad de transmisión de 2.5 Gb/s, como base (módulo de transporte de información de la Jerarquía Digital Síncrona, *Synchronous Digital Hierarchy* SDH, formado a partir de la multiplexación de 16 estructuras básicas STM-1 [7]).

Para obtener velocidades superiores de $M \times 2.5$ Gb/s se entrelazan M señales STM-16. La codificación y decodificación se realizan antes y después del entrelazado, respectivamente.

2.1.2.2 Recomendación UIT-T G.975.1

En la Recomendación UIT-T G.975.1 (figura 2.2) se contempla el uso de superfunciones de corrección de errores en recepción para sistemas que utilizan la tecnología de Multiplexación por División de Longitud de onda Densa (*Dense Wavelength Division Multiplexing*, DWDM).

En este caso las señales de transmisión pueden ser STM-16, STM-64, STM-256 (módulos de transporte de SDH formados a partir de la multiplexación de 16, 64 y 256 señales STM-1, respectivamente [7]) y las unidades de multiplexación ópticas ODU (*Optical channel Data Unit*) [4]: ODU-1 (2.5 Gbps), ODU-2 (10Gbps), ODU-3 (40Gbps) y ODU-4 (100Gbps), que permiten el mapeo de cualquier señal (STM, IP, ATM...).

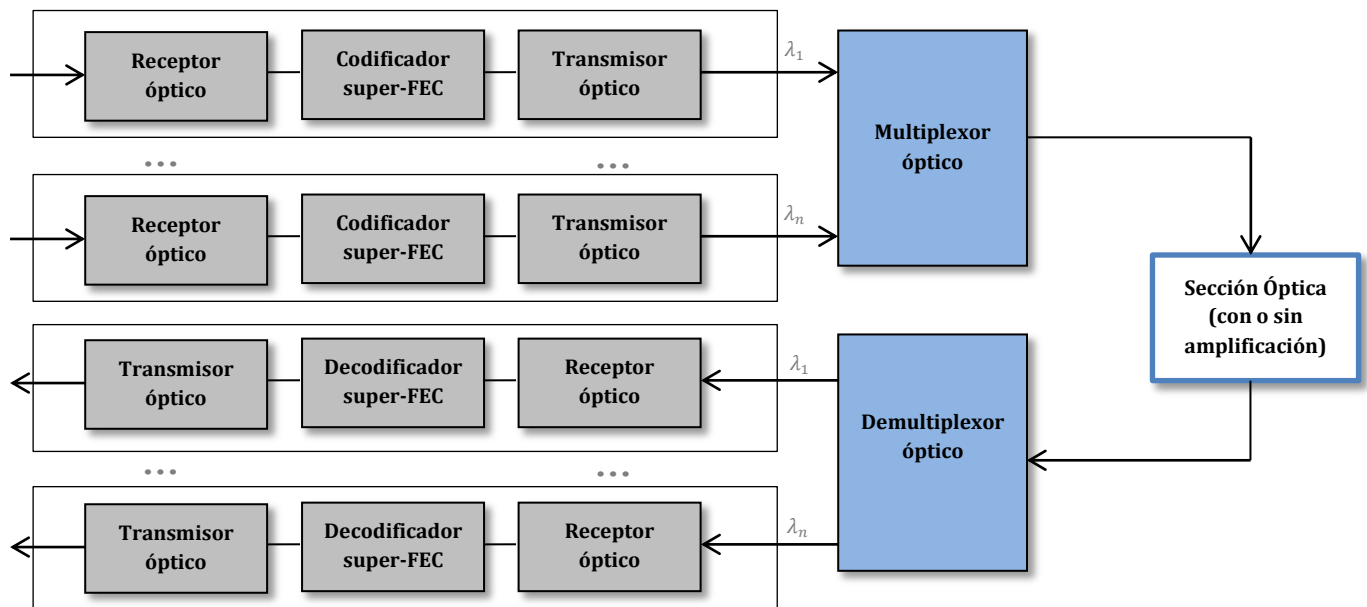


Figura 2.2: Diagrama de bloques de un sistema óptico DWDM. G.975.1

2.2 Evolución de las técnicas FEC

2.2.1 Generaciones de funciones FEC

El aumento de la tasa binaria manteniendo la calidad de la transmisión supone un desafío: en los sistemas de comunicaciones ópticas de muy elevada velocidad la redundancia que se puede añadir está limitada por la evolución y disponibilidad de la circuitería digital y analógica necesaria y los componentes ópticos de alta velocidad.

Las técnicas FEC han mejorado a lo largo de las últimas décadas. En la *figura 2.3* [8] se representa en el eje vertical el producto de la ganancia de codificación y la tasa en Gb/s para una BER de 10^{-15} frente a la evolución temporal en el eje horizontal. Se puede apreciar que el producto mejora a razón de 1.4 veces cada año, debido tanto al progreso de las técnicas FEC como al de las tecnologías de integración de circuitos en gran escala (*Large Scale Integration, LSI*).

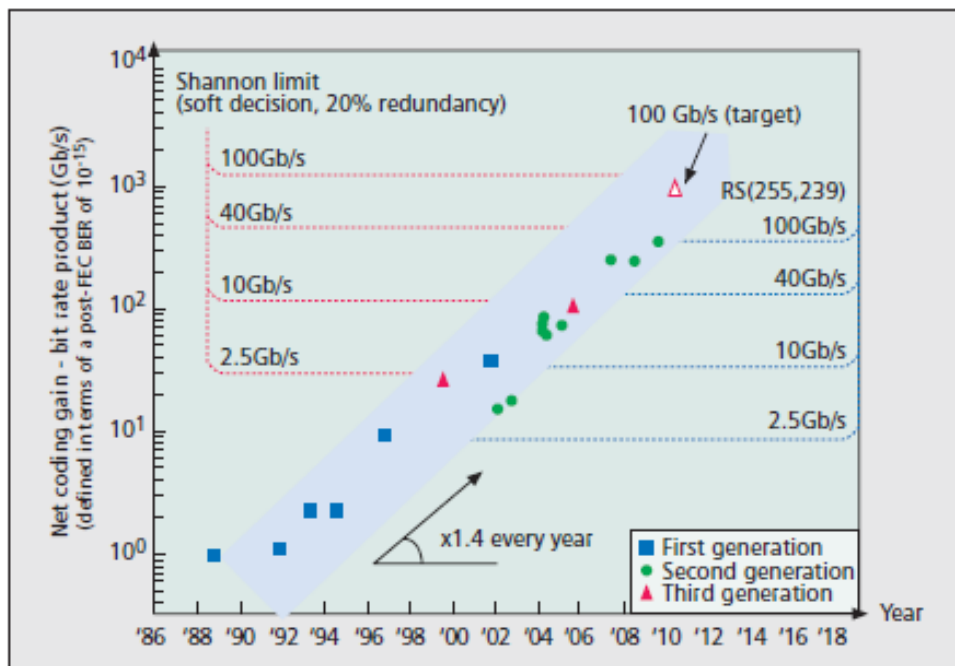


Figura 2.3. Progreso FEC en las últimas décadas [8]

En la gráfica se distinguen tres tipos de nubes de puntos, cada una de ellas asociada a una de las tres generaciones en las que se catalogan dichas técnicas.

2.2.1.1 Inicio de técnicas FEC en comunicaciones ópticas

El potencial de la aplicación de técnicas FEC en los sistemas de comunicaciones ópticas para la transmisión de señales a larga distancia fue descubierto en la década de los 80.

Una de las primeras propuestas prácticas en las que se introdujo FEC en comunicaciones con fibra fue realizada por Grover en 1988 [9]: se aplicó un código Hamming(224,216) en las interfaces de baja velocidad de un sistema de transmisión de 565 Mb/s (se codifican bloques de información de 216 bits resultando una señal codificada de 224). Con una redundancia de 3.7% y una tasa de codificación de 0.96 se consiguió una Ganancia de Codificación Neta (*Net Coding Gain*, NCG) de 2.5 dB para una BER de 10^{-13} .

2.2.1.2 Primera generación FEC

El desarrollo de los códigos FEC de primera generación, caracterizados por emplear decodificación con decisión dura y códigos bloque sistemáticos, se efectúa entre 1987 y 1993.

A mediados de los años 90 la UIT-T prepara la *Recomendación G.975*, la primera en la que aparece un código FEC para proteger la información contra errores en sistemas ópticos de transmisión submarina: el código Reed-Solomon RS(255,239). Se trata de un código bloque, lineal, sistemático, cíclico (si *c* es una palabra perteneciente al código, también pertenecen a él todos los desplazamientos cíclicos de *c*) y no binario (actúa sobre símbolos de 8 bits), con las siguientes propiedades:

- La complejidad del codificador y decodificador presentes en los equipos terminales es reducida.
- La tasa de codificación es $R = 239/255$, es decir, introduce una redundancia del 6.69%. Para una BER de salida de 10^{-12} la NCG es de 5.6 dB.
- La trama a transmitir, compuesta por cuatro filas, se describe en la *Recomendación UIT-T G.709* [4]. Una fila de trama consta de $16 * 255 * 8$ bits (de los cuales 30592 son de información y 2048 son redundancia). Para mejorar la corrección de errores de ráfaga se entrelazan 16 palabras código, pudiendo corregirse ráfagas con una longitud máxima de 1024 bits.

2.2.1.3 Segunda generación FEC

Con la evolución de WDM surge la necesidad de utilizar códigos FEC más potentes que permitan incrementar la capacidad y la distancia de transmisión. En la *Recomendación UIT-T G.975.1* se definen las superfunciones de corrección de errores, la segunda generación FEC.

2. Estado del arte

El objetivo de los códigos de segunda generación, que también utilizan decisión dura, es mejorar la capacidad de corrección del código RS(255,239). En la Recomendación se incluyen diferentes propuestas para aumentar la NCG de la primera generación a través de diferentes estrategias: uso de **códigos concatenados**, **palabras código de mayor longitud** y/o **decodificación iterativa**.

En la Recomendación se proponen los siguientes esquemas:

Esquemas FEC concatenados:

- RS (255,239) + CSOC ($n_0/k_0 = 7/6, J = 8$)
- BCH(3860,3824) + BCH(2040,1930)
- RS(1023,1007) + BCH(2047,1952)
- RS(1901,1855) + código de producto Hamming ampliado(512,502).(510,500)
- Dos códigos BCH concatenados ortogonales
- Dos códigos BCH(1020,988) ampliados y entrelazados

Esquemas FEC no concatenados:

- LDPC
- RS(2720,2550)

En las *tablas 2.1* y *2.2* se resumen las características más relevantes y las prestaciones de los esquemas de codificación de segunda generación:

Esquema súper FEC	Descripción del esquema	Prestaciones	Redundancia
RS (255,239) + CSOC($n_0/k_0 7/6, J = 8$)	Formado por un código exterior RS y un código interior convolucional autoortogonal (CSOC), que permite decodificación iterativa.	NCG de 7.95 dB con BER 10^{-12} y 3 iteraciones.	24.48%
BCH(3860,3824) + BCH(2040,1930)	Compuesto por dos códigos BCH concatenados (un BCH(3860,3824) exterior y un BCH(2040,1930) interior). Permite decodificación iterativa para mejorar la capacidad de corrección.	NCG de 7.98 dB con BER 10^{-12} y decodificación en tres etapas (depende del número de iteraciones).	6.69% (la misma que el código RS(255,239) de UIT-T G.975).
RS(1023,1007) + BCH(2047,1952)	Compuesto por dos códigos entrelazados: RS(1023,1007) exterior y BCH(2047,1952) interior. Se obtienen buenos resultados con una o dos iteraciones.	NCG de 7.63 dB con una BER de 10^{-12} .	6.69% (igual que en UIT-T G.709 y UIT-T G.975).

2. Estado del arte

RS(1901,1855) + producto Hamming ampliado (512,502).(510,500)	Código exterior RS(1901,1855) y código interior de producto Hamming ampliado (512,502).(510,500). Permite decodificación iterativa y con decisión programable con cuantificación de dos o más bits en los códigos de producto.	NCG de 8.4 dB para una BER de 10^{-12} (decodificador con cuantificación de dos bits en recepción, y 8 iteraciones con decisión programable en entrada y salida).	6.69%
Dos BCH concatenados ortogonales	Dos códigos BCH entrelazados ortogonalmente. En la Recomendación se incluyen tres parametrizaciones del esquema: con tara de 7%, 11% y 25%.	Para una BER de 10^{-12} , NCG de 7.53 dB con una tara de 7%, 8.20 dB para una tara de 11% y 9.04 dB con tara del 25% (con decodificación iterativa en cinco etapas).	7%, 11% ó 25%.
Dos BCH(1020,988) ampliados y entrelazados	Dos códigos BCH(1020,988) ampliados y entrelazados, con bloques de la misma longitud. Se realiza decodificación iterativa. La función de entrelazado es sencilla y los polinomios BCH de grado bajo, lo que permite que se puedan realizar fácilmente entre 12 y 20 iteraciones.	NCG de 8.63 dB para una BER de $7 \cdot 10^{-14}$ con 10 iteraciones (mayor a medida que se aumenta el número de iteraciones).	7%

Tabla 2.1: Prestaciones de los esquemas FEC de segunda generación G.975.1 concatenados.

Esquema super FEC	Descripción del esquema	Prestaciones	Redundancia
LDPC	El código es binario y sistemático. La longitud de las palabras código es de 32640 bits, igual que la longitud del código estándar entrelazado RS(255,239). De ellos, coincidiendo con la cabida útil de una trama FEC estándar, 30592 son de información.	NCG de 7.1 dB para una BER de 10^{-12} .	6.69%, (misma que el código RS(255,239)).
RS(2720,2550)	El código es bloque, sistemático, con símbolos de 12 bits y una longitud total de 30592 bits.	NCG de 7.06 dB para una BER de 10^{-12} .	7%

Tabla 2.2: Prestaciones de los esquemas FEC de segunda generación G.975.1 no concatenados.

2.2.1.4 Tercera generación FEC

Las tasas de transmisión de 100Gb/s y superiores requieren el uso de esquemas de codificación que proporcionen una ganancia de codificación mayor a la que ofrecen los códigos FEC de primera y segunda generación, lo que evidencia la necesidad de desarrollar funciones FEC más potentes.

2. Estado del arte

En un principio se proponen técnicas que emplean **decisión dura** (*Hard Decision FEC*, **HD-FEC**) para tasas de 40 Gb/s y 100 Gb/s con los que se ha demostrado que se puede obtener una NCG de aproximadamente 9 dB con una redundancia de 7%, e incluso NCG más elevadas a costa de utilizar redundancias mayores [10].

Como consecuencia de la evolución de los conversores analógico-digitales y los algoritmos avanzados de procesamiento de señal en DSPs (*Digital Signal Processors*), los esquemas FEC con **decisión blanda** (*Soft Decision FEC*, **SD-FEC**) son ya una realidad. Actualmente es mucho más sencillo incorporar en el hardware del receptor decisión blanda, que permite mejorar drásticamente la capacidad de corrección de errores: a costa de incrementar la complejidad, las técnicas SD-FEC proporcionan una ganancia de codificación más elevada que las técnicas HD-FEC.

Los orígenes de las técnicas SD-FEC en comunicaciones ópticas se remontan a 1999, año en el que se realiza la primera demostración experimental de esquemas FEC que utilizan decisión blanda [11]. Con el esquema concatenado propuesto, formado por un código RS(255,239) y un código convolucional de tasa 1/2, se consiguieron unas NCG en torno a 10 dB en diversos escenarios a una tasa de transmisión de 2.5 Gb/s, a costa de añadir una redundancia de 113%.

En 2003 se propuso un esquema FEC que proporciona una NCG DE 10.1 dB a una tasa de 12.4 Gb/s con una redundancia de 24.6% [12] utilizando un turbo código y decisión blanda de 3 bits. La propuesta fue emulada experimentalmente mediante *hardware*, pero no integrada en un circuito de aplicación específica debido a la dificultad de implementación con la tecnología disponible en el momento. Hasta 2006 no se presenta el primer esquema integrado: con un turbo código, a una tasa de 12.4 Gb/s y con una redundancia de 23.6%, se consigue una NCG de 10.1 dB para una BER de 10^{-13} [13].

Desde entonces los esquemas de codificación FEC avanzados de tercera generación han evolucionado muy rápidamente, y parece evidente que las técnicas SD-FEC formarán parte de las redes ópticas del futuro. Con el esquema se ha de conseguir una NCG mayor a 10 dB, para una tasa de 100 Gb/s y una BER de 10^{-15} . Aunque el desarrollo en el ámbito de estas técnicas es muy activo, los avances no se han consolidado todavía: el objetivo más inmediato es encontrar una solución óptima para la transmisión de 100 Gb/s y 400 Gb/s.

El *Optical Internetworking Forum* (OIF) analiza en [14] la máxima redundancia que puede ser añadida con los códigos correctores de errores en sistemas de transporte de comunicaciones

ópticas DWDM de ultra larga distancia, a una tasa de 100 Gb/s y con modulación DP-QPSK (*Dual Polarization-Quadrature Phase Shift Keying*), una modulación digital que utiliza dos polarizaciones ortogonales, con modulación QPSK en cada polarización. Siguiendo las recomendaciones del OIF, el esquema ha de tener una **redundancia inferior a 20%**.

2.2.2 Códigos LDPC para tercera generación FEC

A pesar de que la determinación de qué tipo de esquema es el más adecuado es un debate que se encuentra hoy en día abierto, la inmensa mayoría de líneas de investigación actuales se orientan hacia los códigos LDPC y las técnicas SD-FEC: se ha demostrado que con un diseño apropiado pueden proporcionar una excelente ganancia de codificación en la región de interés de las comunicaciones ópticas.

Hasta el momento se han propuesto esquemas muy diversos y numerosas variaciones sobre los mismos para los sistemas de nueva generación. Por ejemplo, Chang presenta un código LDPC *quasi cíclico* (QC-LDPC) con una NCG de 11.3 dB y una redundancia de 20% sin *error floor* en BER superiores a 10^{-15} , verificado en un emulador implementado en FPGA [15]; Dave un turbo código con una NCG de 11 dB para una BER de 10^{-15} y una redundancia de 15% [16] y Miyata un esquema FEC con triple concatenación. En el artículo [17] propone un esquema basado en un código LDPC concatenado con esquemas súper FEC de segunda generación con una redundancia total de 20.5%, con el que se obtiene una NCG de 10.8 dB para una BER 10^{-15} .

La mayoría de las propuestas de códigos LDPC realizadas hasta hace unos años se basan en códigos bloque, cuyas prestaciones mejoran en la medida en que se utilizan códigos con longitudes mayores. Por otro lado incrementar la longitud dificulta el diseño del codificador y el decodificador al existir un número de conexiones más elevado, y se aumentan los requisitos de espacio y el consumo de potencia. A pesar de que el uso de códigos LDPC-CC no está todavía afianzado, se conoce que tienen un gran potencial y que poseen diversas propiedades que los hacen idóneos para las aplicaciones de comunicaciones ópticas de alta velocidad.

Existe un especial interés en torno a los códigos QC-LDPC regulares, ya que poseen buenas prestaciones en la región de *error floor* y permiten una implementación *hardware* eficiente y de baja complejidad [18]. La matriz de chequeo de paridad de un código QC-LDPC regular es:

$$H = \begin{bmatrix} I & I & I & \dots & I \\ I & p^{S[1]} & p^{S[2]} & \dots & p^{S[c-1]} \\ I & p^{2S[1]} & p^{2S[2]} & \dots & p^{2S[c-1]} \\ \dots & \dots & \dots & \dots & \dots \\ I & p^{(r-1)S[1]} & p^{(r-1)S[2]} & \dots & p^{(r-1)S[c-1]} \end{bmatrix} \quad (2.1)$$

Donde I es la matriz identidad $[p] \times [p]$ (siendo p un número primo), r y c representan el número de bloques de filas y columnas, respectivamente, y las matrices P son permutaciones $[p] \times [p]$:

$$P = (p_{ij})_{p \times p}, \quad p_{i,i+1} = p_{p,1} = 1, \text{ y } 0 \text{ en otro caso.} \quad (2.2)$$

La construcción de un código LDPC se define mediante el proceso de diseño de su matriz de chequeo de paridad. Tampoco la problemática de la construcción de la matriz de chequeo está resuelta en el caso de los códigos QC-LDPC. En [19], artículo tomado como referencia para evaluar las prestaciones de los códigos simulados en este proyecto, Che y Wang presentan un nuevo método para la construcción de la matriz H con el que proponen dos códigos QC-LDPC(32640,30601) y QC-LDPC(12240,10201). En la *figura 2.4* se representa la BER en función de la E_b/N_0 de dichos códigos, así como una comparación con los de primera y segunda generación (RS(255,239) y G.975.1 I.7 – dos códigos BCH concatenados de forma ortogonal –, respectivamente).

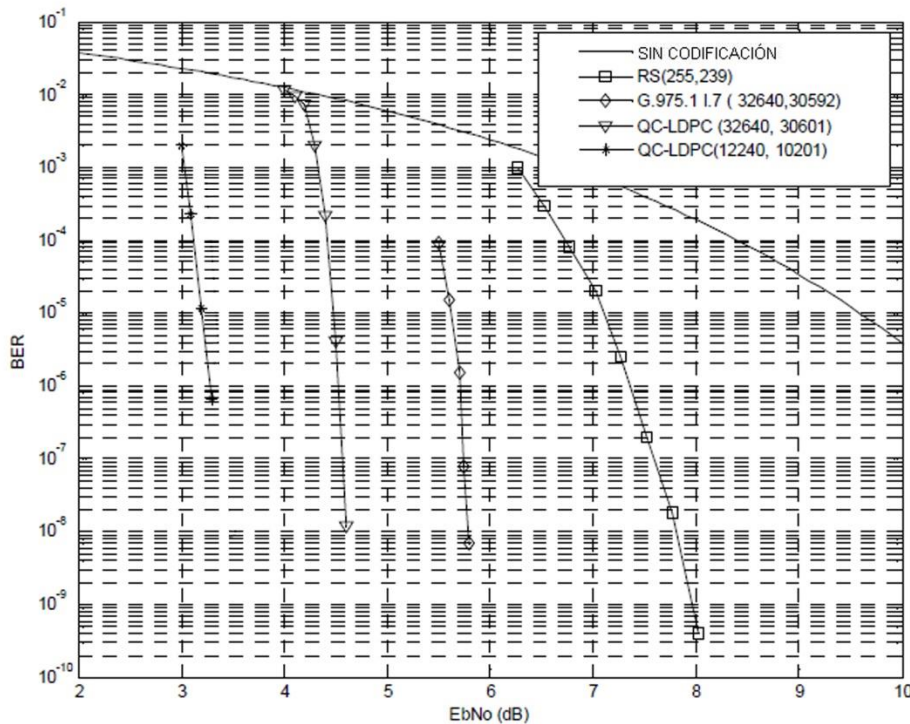


Figura 2.4 Comparativa BER en función de E_b/N_0 . Transmisión BPSK, canal AWGN [19]

2. Estado del arte

Djordjevic analiza en [18] las prestaciones de los códigos QC-LDPC irregulares, con los que se pueden conseguir unas mayores ganancias de codificación, aunque este tipo de códigos no son óptimos desde el punto de vista de implementación física del decodificador. Por otro lado también hay un creciente interés en torno a las variantes que pueden originarse a partir de la reordenación de la matriz de chequeo de paridad de los códigos QC-LDPC, permitiendo una mayor flexibilidad en el diseño: los códigos *staircase* [20] y *spatially coupled* (SC-LDPC).

Este proyecto se centra en los códigos LDPC con **Acoplamiento Espacial** o *Spatially Coupled*. Aunque no existe una tendencia convergente a la hora de predecir la tecnología FEC que será utilizada, el análisis y estudio de estos códigos en concreto se encuentra en auge y son susceptibles de ser utilizados en futuros estándares de comunicaciones ópticas.

La estructura de la matriz de chequeo de paridad (*figura 2.5*), en la que las conexiones se encuentran restringidas a un bloque diagonal, facilita una implementación física eficiente del decodificador: gracias a la disposición localizada de los “unos” se simplifica el diseño *hardware*. Además es posible realizar **decodificación en ventanas**, es decir, se puede aplicar el algoritmo de decodificación iterativo sobre submatrices de \mathbf{H} y subconjuntos de las palabras recibidas. La ventana se va desplazando a medida que se decodifican un grupo de símbolos dentro de ella y se reciben los siguientes [2]. En el capítulo tercero se describen en detalle el procedimiento de construcción de los códigos SC-LDPC y el algoritmo de decodificación en ventanas.

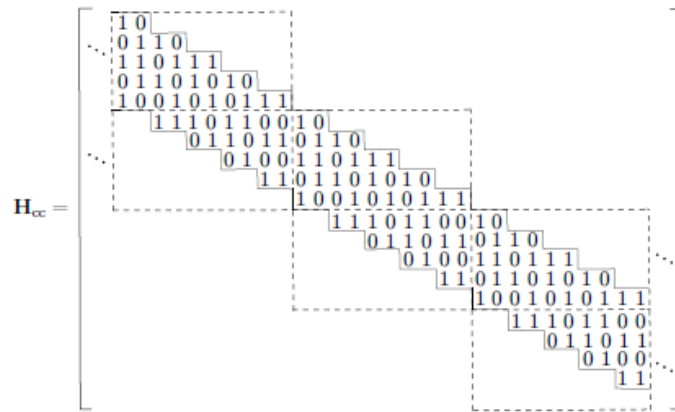


Figura 2.5 Estructura de la matriz de chequeo de un código SC-LDPC

2. Estado del arte

En [2] se plantea que existen todavía multitud de temas sin resolver en relación con el diseño de codificadores y decodificadores SC-LDPC:

- Se ha de analizar cómo afecta la **variación de los diferentes parámetros de construcción del código**, expuestos en el capítulo siguiente. También se ha de implementar un **algoritmo de decodificación enventanada** eficiente, eligiendo un tamaño de ventana óptimo y una condición de parada para determinar cuándo se han realizado suficientes iteraciones con el algoritmo de decodificación en cada ventana y ésta ha de desplazarse. Otra cuestión de interés son las técnicas de perforación (eliminación de dígitos de las palabras código) para obtener tasas elevadas y flexibles apropiadas para ser utilizadas en estándares, y el estudio de otros modos de mitigar la pérdida de tasa (también explicada en el capítulo posterior) derivada de las terminaciones que se introducen en el código.
- Ha de realizarse una **comparativa entre la complejidad y prestaciones** de los códigos LDPC-BC y SC-LDPC: se debe determinar si los códigos SC-LDPC con decodificación en ventanas proporcionan unas mejores prestaciones que los códigos bloque en términos de probabilidad de error. Para que la comparación se realice de modo justo deben igualarse la latencia, necesidades de memoria y complejidad computacional en ambos casos: el tamaño de los códigos bloque ha de ser equivalente al tamaño de la ventana. Los códigos a comparar han de tener grados semejantes, con el mismo número de unos por fila y columna.

3

Codificación LDPC

Contenidos

3.1 Códigos bloque LDPC.....	31
3.1.1 Representación gráfica de la matriz de chequeo de paridad	
3.1.2 Algoritmo de decodificación Belief Propagation	
3.2 Códigos SC-LDPC.....	39
3.2.1 Estructura de los códigos SC-LDPC	
3.2.2 SC-LDPC basado en protographs	
3.2.3 Construcción de la matriz de chequeo de paridad	
3.2.4 Decodificación en ventanas	

3. Análisis Teórico

Una vez expuesta la evolución de las técnicas FEC en los sistemas de comunicaciones ópticas a lo largo de las últimas décadas y puesta en evidencia la necesidad de estandarizar códigos con una mayor capacidad correctora de errores, en el presente capítulo se realiza un análisis teórico de uno de los numerosos tipos de códigos que hasta el momento han sido propuestos para ser utilizados en las redes de nueva generación y en los que se centra este trabajo: los códigos SC-LDPC.

En la *tabla 3.1* se resumen las principales características que presentan los códigos LDPC en función de la distribución de los grados. También se realiza en la misma una comparativa con las características de los códigos SC-LDPC, poniéndose de manifiesto que permiten combinar las ventajas de los códigos LDPC-BC regulares e irregulares [2]:

	Códigos bloque LDPC regulares	Códigos bloque LDPC irregulares	Códigos SC-LDPC
Distribución de los grados	Regular: mismo número de '1' por fila y columna en H	Irregular: distinto número de '1' por fila y/o columna en H	<i>Quasi</i> -regular: únicamente presentan irregularidades estructuradas en los extremos
Prestaciones curva BER	No alcanzan el límite de Shannon con algoritmos de decodificación iterativos	Próximas al límite de Shannon con algoritmos de decodificación iterativos	Próximas al límite de Shannon con algoritmos de decodificación iterativos
Error Floor	No presentan fenómeno de <i>error floor</i>	Son necesarias longitudes de palabra muy elevadas para mantener bajo el <i>error floor</i>	Comportamiento similar a códigos regulares en términos de <i>error floor</i>
Implementación de codificador/decodificador	Sencilla, gracias a la distribución regular de los grados	Compleja, debido a la distribución irregular de los grados y las largas longitudes de palabra requeridas	Sencilla, gracias a la distribución <i>quasi</i> -regular de los grados y la estructura de la matriz de chequeo. Se puede alcanzar un compromiso entre prestaciones y latencia con un algoritmo de decodificación en ventanas

Tabla 3.1: Características de códigos LDPC en función de la distribución de los grados

En primer lugar se realiza una revisión de los códigos bloque LDPC, tratando la representación gráfica de su matriz de chequeo de paridad y el algoritmo de decodificación iterativo *Belief Propagation*. Este análisis previo resulta imprescindible para abordar posteriormente los códigos SC-LDPC: se presenta la estructura de su matriz de chequeo y la construcción basada en unos grafos llamados *protographs*. Finalmente, se aborda la arquitectura del decodificador en ventanas.

3.1 CÓDIGOS BLOQUE LDPC

3.1.1 Representación gráfica de la matriz de chequeo de paridad

La matriz de chequeo de paridad de los códigos LDPC puede representarse gráficamente a través de grafos bipartitos llamados **grafos de Tanner**. La importancia de esta herramienta gráfica reside en que es la base de los algoritmos eficientes de decodificación iterativa [3] [21].

Con el **grafo de Tanner** se representa la estructura de un código LDPC-BC. Existen dos conjuntos o dos tipos de nodos: uno asociado a los bits de la palabra recibida y otro a las comprobaciones de paridad. En total hay n nodos del primer tipo, conocidos como nodos **variables**, y $n-k$ nodos del segundo tipo, llamados nodos de **chequeo**. El grafo se construye a partir de la matriz \mathbf{H} : un nodo de chequeo NC_i está conectado a un nodo variable NV_j si el elemento (i,j) de la matriz \mathbf{H} es un uno.

En la *figura 3.1* se muestra el grafo de Tanner para un código binario, con 8 nodos variables y 4 nodos de chequeo:

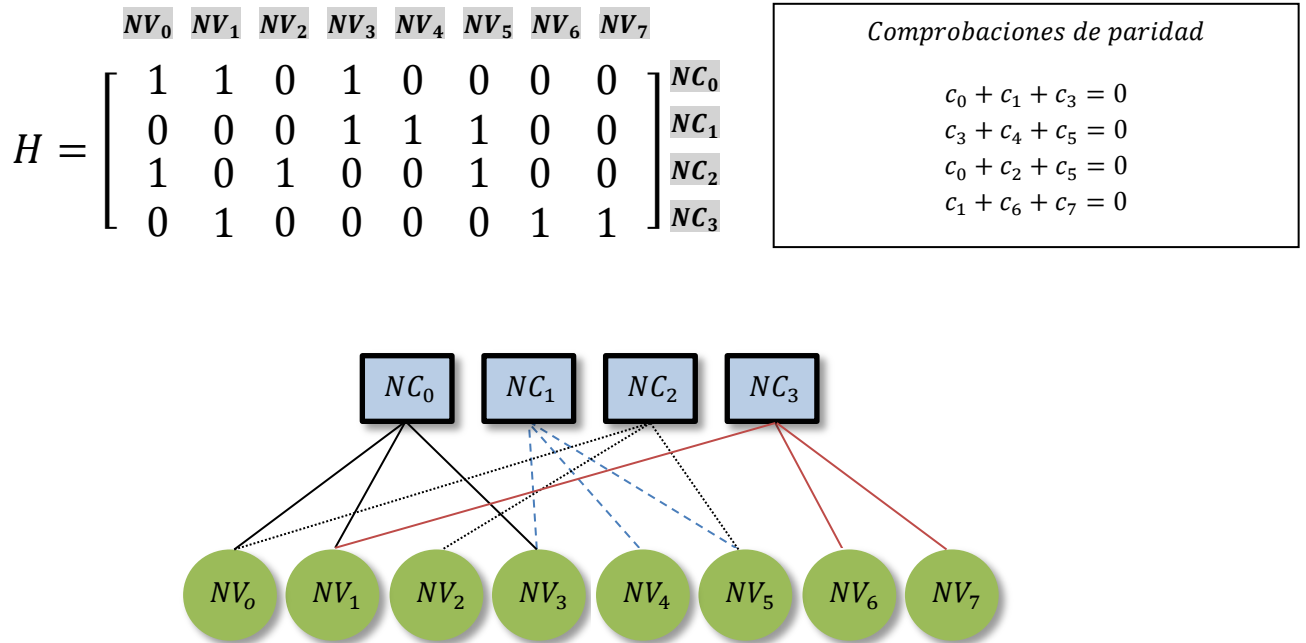


Figura 3.1: Grafo de Tanner construido a partir de H

3.1.2 Algoritmo de decodificación Belief Propagation

En la *figura 3.2* se ilustra el diagrama de bloques del proceso de codificación y decodificación:

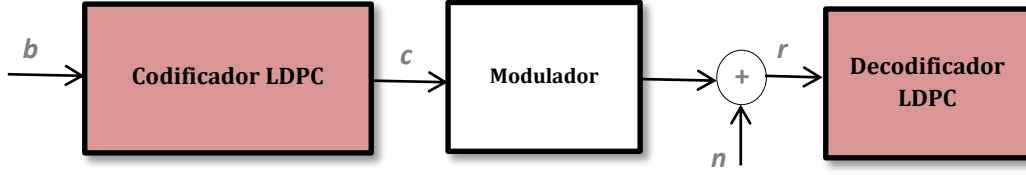


Figura 3.2: Diagrama de bloques de codificación/decodificación LDPC

El decodificador MAP de bloque minimiza la probabilidad de error de bloque $P(\hat{\mathbf{c}} \neq \mathbf{c}^*)$ y proporciona a su salida la palabra $\hat{\mathbf{c}}$ perteneciente al código \mathbf{C} (satisface la ecuación de comprobación de paridad $\mathbf{c}\mathbf{H}^T = 0$) que maximiza la *probabilidad a posteriori*:

$$\hat{\mathbf{c}} = \arg \max_{\mathbf{c} \in \mathbf{C}} p(\mathbf{c}/\mathbf{r}, \mathbf{c}\mathbf{H}^T = 0) \quad (3.1)$$

El decodificador MAP de bit minimiza la probabilidad de error de bit $\frac{1}{n} \sum_{i=1}^n P(\hat{c}_i \neq c_i)$ y estima el i -ésimo bit de la palabra código:

$$\hat{c}_i = \arg \max_{c_i \in \{0,1\}} p(c_i|\mathbf{r}) \quad (3.2)$$

La complejidad del decodificador MAP es en ambos casos $O(2^n)$, es decir, crece exponencialmente con la longitud de la palabra código haciendo que su uso con códigos LDPC sea inviable. El éxito de los códigos LDPC surge de su combinación con una instancia de un algoritmo de decodificación iterativo de paso de mensajes: el algoritmo **Belief Propagation** (BP). A pesar de ser un método subóptimo de decodificación el algoritmo BP es computacionalmente mucho más eficiente. Su complejidad crece linealmente con la longitud de la palabra código, haciendo que sea posible trabajar con matrices de decenas de miles de bits.

El decodificador BP obtiene una aproximación de la probabilidad marginal *a posteriori* de cada uno de los bits, satisfaciendo únicamente las ecuaciones de comprobación en las que aparece dicho bit. Tanto la calidad de la aproximación como la carga computacional dependen directamente de la densidad de unos de la matriz \mathbf{H} .

3. Análisis Teórico

3.1.2.1 Paso de mensajes

El algoritmo de decodificación puede ser visto como una instancia de un algoritmo de paso de mensajes. En la *figura 3.3* se representa cómo el algoritmo opera sobre el grafo: cada iteración se corresponde con un intercambio de mensajes a través del grafo de Tanner entre los dos conjuntos de nodos (variables y de chequeo).

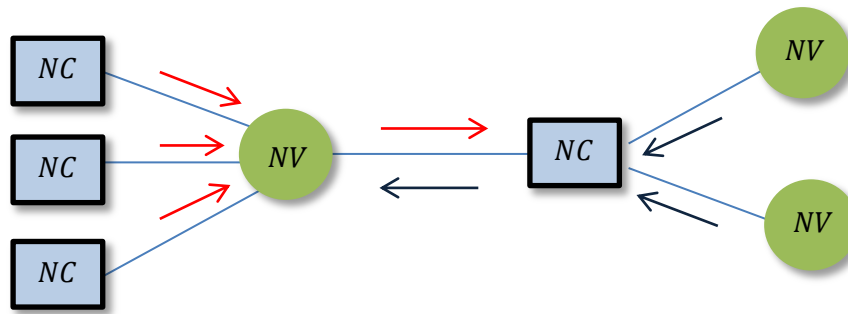


Figura 3.3: Intercambio de mensajes, en algoritmo de paso de mensajes

En cada iteración se efectúan dos pasos, en los que ambos conjuntos de nodos se intercambian mensajes:

Actualización Horizontal: se actualizan los nodos de comprobación a partir de los mensajes generados por los nodos variables vecinos.

Actualización Vertical: se actualizan los nodos variables a partir de los mensajes recibidos de los nodos de chequeo.

De modo general el algoritmo utiliza los mensajes, que se propagan a través del árbol derivado del grafo de Tanner, para calcular la *probabilidad a posteriori* de cada uno de los bits. En ausencia de ciclos el algoritmo proporciona probabilidades exactas, pero la existencia de ciclos hace que se obtengan aproximaciones de las probabilidades. En este proyecto se implementa el algoritmo de decodificación formulado en términos del **logaritmo del cociente de verosimilitudes** (*Log-Likelihood Ratio*, LLR), aplicable a códigos binarios [21].

3.1.2.2 Notación

La matriz H es dispersa, por lo que resulta eficiente utilizar una notación basada en identificar los elementos de H que son no nulos.

3. Análisis Teórico

Elementos no nulos en la m-ésima fila

El conjunto de nodos variable que participan en una ecuación de comprobación z_m (elementos no nulos en la m -ésima fila de H) es:

$$N_m = \{n: H_{m,n} = 1\} \quad (3.3)$$

En consecuencia se puede escribir la m -ésima ecuación de comprobación:

$$z_m = \sum_{n \in N_m} c_n \quad (3.4)$$

El conjunto de bits involucrados en una ecuación de comprobación z_m excepto el bit n -ésimo:

$$N_{m,n} = N_m \setminus n \quad (3.5)$$

Elementos no nulos en la n-ésima columna

El conjunto de ecuaciones de comprobación en las que interviene un determinado nodo variable c_n (elementos no nulos en la n -ésima columna de H) se define como:

$$M_n = \{m: H_{m,n} = 1\} \quad (3.6)$$

Las ecuaciones en las que participa el bit c_n excepto la ecuación m -ésima es:

$$M_{n,m} = M_n \setminus m \quad (3.7)$$

3.1.2.3 Algoritmo de cociente de verosimilitudes

El decodificador realiza una estimación del logaritmo de la razón de verosimilitudes *a posteriori* (LLR) de cada bit transmitido:

$$\lambda(c_n|r) = \log \frac{p(c_n = 1|r)}{p(c_n = 0|r)} = \log \frac{p(c_n = 1|r_n, \{r_i, i \neq n\})}{p(c_n = 0|r_n, \{r_i, i \neq n\})} \quad (3.8)$$

donde r_n es el n -ésimo bit de la palabra recibida y $\{r_i, i \neq n\}$ son el resto de bits. Se decide $\hat{c}_n = 1$ si $\lambda(c_n|r) > 0$ y $\hat{c}_n = 0$ en otro caso.

3. Análisis Teórico

Aplicando la regla de Bayes (Apéndice B.1) en el numerador:

$$\begin{aligned} p(c_n = 1 | r_n, \{r_i, i \neq n\}) &= \frac{p(r_n, c_n = 1, \{r_i, i \neq n\})}{p(r_n, \{r_i, i \neq n\})} \\ &= \frac{p(r_n | c_n = 1, \{r_i, i \neq n\})p(c_n = 1, \{r_i, i \neq n\})}{p(r_n | \{r_i, i \neq n\})p(\{r_i, i \neq n\})} \end{aligned} \quad (3.9)$$

Suponiendo c_n y r_n independientes de $\{r_i, i \neq n\}$ y aplicando Bayes de nuevo:

$$\begin{aligned} p(c_n = 1 | r_n, \{r_i, i \neq n\}) &= \frac{p(r_n | c_n = 1)p(c_n = 1, \{r_i, i \neq n\})}{p(r_n | \{r_i, i \neq n\})p(\{r_i, i \neq n\})} = \\ &= \frac{p(r_n | c_n = 1)p(c_n = 1 | \{r_i, i \neq n\})}{p(r_n | \{r_i, i \neq n\})} \end{aligned} \quad (3.10)$$

Procediendo del mismo modo en el denominador se obtiene:

$$p(c_n = 0 | r_n, \{r_i, i \neq n\}) = \frac{p(r_n | c_n = 0)p(c_n = 0 | \{r_i, i \neq n\})}{p(r_n | \{r_i, i \neq n\})} \quad (3.11)$$

Podemos reescribir por tanto el cociente de verosimilitudes:

$$\begin{aligned} \lambda(c_n | r) &= \log \frac{p(c_n = 1 | r)}{p(c_n = 0 | r)} = \log \frac{p(r_n | c_n = 1)p(c_n = 1 | \{r_i, i \neq n\})p(r_n | \{r_i, i \neq n\})}{p(r_n | c_n = 0)p(c_n = 0 | \{r_i, i \neq n\})p(r_n | \{r_i, i \neq n\})} = \\ &= \log \frac{p(r_n | c_n = 1)}{p(r_n | c_n = 0)} + \log \frac{p(c_n = 1 | \{r_i, i \neq n\})}{p(c_n = 0 | \{r_i, i \neq n\})} \end{aligned} \quad (3.12)$$

El cociente se puede expresar finalmente como la suma de dos términos:

Término intrínseco $\left(\log \frac{p(r_n | c_n = 1)}{p(r_n | c_n = 0)} \right)$: en este término influye únicamente el bit n -ésimo de la palabra recibida.

Término extrínseco $\left(\log \frac{p(c_n = 1 | \{r_i, i \neq n\})}{p(c_n = 0 | \{r_i, i \neq n\})} \right)$: en este término contribuyen el resto de observaciones $\{r_i, i \neq n\}$

3. Análisis Teórico

A continuación se reescribe el término extrínseco en términos de las ecuaciones de comprobación. Siendo $z_{m,n}$ el resultado de la m -ésima ecuación de comprobación, excluyendo el n -ésimo nodo variable:

$$z_{m,n} = \sum_{i \in N_{m,n}} c_i \quad (3.13)$$

Si $c_n = 1$, $z_{m,n} + c_n = 0$ para todas las ecuaciones M_n . La probabilidad de que c_n sea uno es entonces $p(c_n = 1) = p(z_{m,n} = 1, m \in M_n)$.

El término extrínseco puede expresarse por tanto del siguiente modo:

$$\log \frac{p(c_n = 1 | \{r_i, i \neq n\})}{p(c_n = 0 | \{r_i, i \neq n\})} = \log \frac{p(z_{m,n} = 1, m \in M_n | \{r_i, i \neq n\})}{p(z_{m,n} = 0, m \in M_n | \{r_i, i \neq n\})} \quad (3.14)$$

Asumimos que el grafo no tiene ciclos. En consecuencia el conjunto de bits asociados con $z_{m,n}$ son independientes de los asociados con $z_{m',n'}$:

$$\begin{aligned} \log \frac{p(z_{m,n} = 1, m \in M_n | \{r_i, i \neq n\})}{p(z_{m,n} = 0, m \in M_n | \{r_i, i \neq n\})} &= \log \frac{\prod_{m \in M_n} p(z_{m,n} = 1 | \{r_i, i \neq n\})}{\prod_{m \in M_n} p(z_{m,n} = 0 | \{r_i, i \neq n\})} = \\ &= \sum_{m \in M_n} \log \frac{p(z_{m,n} = 1 | \{r_i, i \neq n\})}{p(z_{m,n} = 0 | \{r_i, i \neq n\})} = \sum_{m \in M_n} \log \frac{p(\sum_{j \in N_{m,n}} c_j = 1 | \{r_i, i \neq n\})}{p(\sum_{j \in N_{m,n}} c_j = 0 | \{r_i, i \neq n\})} \end{aligned} \quad (3.15)$$

Renombramos el término $\log \frac{p(z_{m,n}=1 | \{r_i, i \neq n\})}{p(z_{m,n}=0 | \{r_i, i \neq n\})} = \lambda(z_{m,n} = 1 | \{r_i, i \neq n\})$:

$$\log \frac{p(z_{m,n} = 1, m \in M_n | \{r_i, i \neq n\})}{p(z_{m,n} = 0, m \in M_n | \{r_i, i \neq n\})} = \sum_{m \in M_n} \lambda \left(\sum_{j \in N_{m,n}} c_j | \{r_i, i \neq n\} \right) \quad (3.16)$$

Suponiendo independencia y aplicando la regla de la tangente (Apéndice B.2):

$$\begin{aligned} \log \frac{p(c_n = 1 | \{r_i, i \neq n\})}{p(c_n = 0 | \{r_i, i \neq n\})} &= \sum_{m \in M_n} \lambda \left(\sum_{j \in N_{m,n}} c_j | \{r_i, i \neq n\} \right) = \\ &= -2 \sum_{m \in M_n} \tanh^{-1} \left(\prod_{j \in N_{m,n}} \tanh \left(-\frac{1}{2} \lambda(c_j | \{r_i, i \neq n\}) \right) \right) \end{aligned} \quad (3.17)$$

3. Análisis Teórico

El mensaje enviado desde el nodo de comprobación m -ésimo hacia el nodo variable n -ésimo se relaciona con el siguiente término:

$$\eta_{m,n} = -2 \tanh^{-1} \left(\prod_{j \in N_{m,n}} \tanh \left(-\frac{1}{2} \lambda(c_j | \{r_i, i \neq n\}) \right) \right) \quad (3.18)$$

Cada nodo variable computa el cociente de verosimilitudes sumando todos los mensajes recibidos desde los nodos de comprobación y añadiendo el término intrínseco:

$$\lambda(c_n | r) = \log \frac{p(r_n | c_n=1)}{p(r_n | c_n=0)} + \sum_{m \in M_n} \eta_{m,n} \quad (3.19)$$

El algoritmo se resume en los siguientes pasos:

1. Inicialización:

$$\eta_{m,n}^{[0]} = 0, \text{ para todos los elementos } (m,n) \text{ con } H(m,n) = 1$$

$$\lambda_n^{[0]} = \log \frac{p(r_n | c_n=1)}{p(r_n | c_n=0)} \quad (3.20)$$

2. Actualización horizontal: se calculan los mensajes con información extrínseca que los nodos de comprobación m -ésimos envían a los nodos variables n -ésimos.

Para cada elemento (m,n) con $H(m,n) = 1$, en la iteración l -ésima:

$$\eta_{m,n}^{[l]} = -2 \tanh^{-1} \left(\prod_{j \in N_{m,n}} \tanh \left(-\frac{1}{2} (\lambda_j^{[l-1]} - \eta_{m,j}^{[l-1]}) \right) \right) \quad (3.21)$$

Las verosimilitudes $\lambda(c_n | r)$ contienen también la información asociada al término intrínseco. Para evitar que se produzca un *offset* en los mensajes $\eta_{m,n}$, en el algoritmo se elimina del mensaje recibido por el nodo variable n -ésimo el mensaje enviado por el nodo de chequeo m -ésimo en la iteración anterior.

3. Análisis Teórico

3. Actualización vertical: se calculan los mensajes que los nodos variables envían a todos los nodos de chequeo vecinos en la siguiente iteración: el cociente de verosimilitudes.

para cada nodo variable $n, n = 1, 2 \dots N$, en la iteración l – ésima:

$$\lambda_n^{[l]} = \log \frac{p(r_n | c_n = 1)}{p(r_n | c_n = 0)} + \sum_{m \in M_n} \eta_{m,n}^{[l]} \quad (3.22)$$

4. Decisión

$$\lambda_n^{[l]} = \log \frac{\hat{p}(c_n = 1 | r)}{\hat{p}(c_n = 0 | r)} \quad (3.23)$$

Si $\lambda_n^{[l]} > 0$ se decide $\hat{c}_n = 1$, y $\hat{c}_n = 0$ en caso contrario.

A continuación se comprueba si la palabra decodificada cumple todas las ecuaciones de comprobación $\hat{c}H^T = 0$, en cuyo caso el algoritmo finaliza. Si no es así el algoritmo continúa iterando volviendo al paso de actualización horizontal, siempre que l sea inferior al número máximo de iteraciones permitido. De este modo, en cada iteración se recibe información de vecinos más alejados. Si el número de iteraciones ha alcanzado el máximo sin que se cumpla dicha condición se declara un fallo en decodificación.

3.2 CÓDIGOS SC-LDPC

En esta sección se describen las características de los códigos LDPC con **Acoplamiento Espacial** o *Spatially Coupled*, seleccionados por poseer las siguientes propiedades: la estructura de la matriz de chequeo y la distribución *quasi-regular* de los grados de los nodos facilitan el diseño *hardware* de la arquitectura del codificador y decodificador, proporcionan unas prestaciones cercanas al límite de Shannon y su estructura permite emplear un algoritmo de decodificación en ventanas, con el que puede conseguirse una latencia similar a la obtenida con los códigos LDPC-BC.

Finalmente se detalla el método utilizado para construir las matrices de chequeo de paridad y el algoritmo de decodificación en ventanas.

3.2.1 Estructura de los códigos SC-LDPC

Los códigos SC-LDPC pueden ser vistos como un tipo de código LDPC convolucional: la introducción de acoplamiento espacial es equivalente a la introducción de memoria en el proceso de codificación [2]. Un código LDPC convolucional se define a través de una matriz de chequeo de paridad \mathbf{H}_{cc} con baja densidad de unos que se extiende infinitamente a lo largo del tiempo:

$$H_{cc} = \begin{bmatrix} H_{m_s}(t-1) & \ddots & & & & & \\ & H_{m_s-1}(t-1) & \dots & & H_0(t-1) & & \\ & H_{m_s}(t) & H_{m_s-1}(t) & & \dots & H_0(t) & \\ & & H_{m_s}(t+1) & & & \dots & H_0(t+1) \\ & & & \ddots & & & \\ & & & & \ddots & & \end{bmatrix} \quad (3.24)$$

Estos códigos tienen una estructura definida y localizada: los unos de la matriz de chequeo se encuentran dispersos en una región diagonal determinada por los parámetros m_s y v_s , el resto de elementos son nulos.

- m_s : es la *memoria del código*. La región diagonal se compone de un conjunto de $m_s + 1$ submatrices:

$$H_i(t), \quad 0 \leq i \leq m_s, \quad t = 0, 1, 2, \dots \quad (3.25)$$

3. Análisis Teórico

- v_s : es la **longitud de decodificación** asociada. Los elementos no nulos se restringen únicamente a una banda diagonal con una anchura de v_s bits.

La matriz \mathbf{H}_{cc} se extiende infinitamente. Para implementar el código es necesario terminarlo después de \mathbf{L} instantes:

$$H_L = \begin{pmatrix} H_0(1) & & & & \\ H_1(1) & H_0(2) & & & \\ \dots & & \ddots & & \\ H_{m_s}(1) & H_1(2) & \ddots & H_0 & \\ & \vdots & \ddots & H_1(L) & \\ & H_{m_s}(2) & \ddots & \vdots & \\ & & & H_{m_s}(L) & \end{pmatrix} \quad (3.26)$$

Para que la matriz obtenida tenga una baja densidad de unos se ha de cumplir que el número de unos por fila y columna ha de ser muy bajo. El código es regular (\mathbf{J}, \mathbf{K}) si la matriz \mathbf{H}_L tiene \mathbf{J} unos en cada columna y \mathbf{K} unos en cada fila. Se excluyen un conjunto de columnas en los extremos de la matriz: los nodos de chequeo en los extremos del grafo se encuentran conectados a un menor número de nodos variables debido a la terminación del código.

En definitiva, los parámetros que definen la estructura de este tipo de códigos son: los **grados de los nodos** (\mathbf{J}, \mathbf{K}), el número de instantes tras los que se termina el código \mathbf{L} , el **patrón de acoplamiento** y la longitud total del código $\mathbf{n}=\mathbf{ML}$.

3.2.2 SC-LDPC basado en *protographs*

Puede utilizarse un grafo bipartito llamado **protograph** para representar la estructura de la matriz de chequeo de paridad de un código SC-LDPC. Estas representaciones de los códigos convolucionales y de los códigos convolucionales terminados fueron introducidas por Mitchell, Pusane, Zigangirov y Costello en [22] y Lentmaier, Fettweis, Costello y Zigangirov en [23], respectivamente.

Un *protograph* está formado por un conjunto de nodos variables y nodos de comprobación y los enlaces entre ambos. En la *figura 3.4* se incluye el *protograph* de un código LDPC-BC(3,6) de tasa $R = 1 - \frac{J}{K} = \frac{k}{n} = \frac{1}{2}$, con $n_v = 2$ conjuntos de nodos variables de grado 3 y $n_c = n - k = 1$ conjunto de nodos de chequeo de grado 6. La matriz \mathbf{B} que indica el número de conexiones

3. Análisis Teórico

entre los nodos variables y de chequeo, de dimensiones $[n_v - k] \times [n_v]$, se conoce como matriz base. En el ejemplo propuesto la matriz es $B = [3, 3]$.

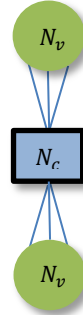


Figura 3.4: Protograph de un LDPC-BC(3,6), $n_v = 2, n_c = 1, R = \frac{1}{2}$

Para formar un código regular LDPC-BC(3,6) con palabras código de longitud M bits se realiza una expansión, de modo que cada nodo N_v representa un total de $M_d = \frac{M}{n_v}$ bits. La expansión se lleva a cabo sustituyendo cada elemento de la matriz base \mathbf{B} por una matriz de dimensiones $[M_d] \times [M_d]$.

Replicación del protograph

Se replica el *protograph* L veces (figura 3.5), obteniendo una secuencia de códigos idénticos con una longitud total de $n = LM$. Estos *protographs* no se relacionan entre sí: cada componente se comporta como un código independiente.

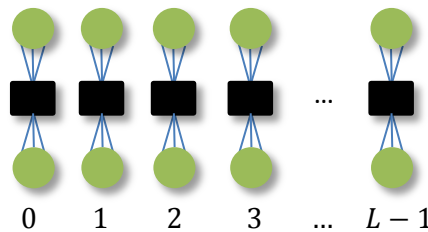


Figura 3.5: Cadena de protographs no acoplados

Acoplamiento de las submatrices

Para hacer el acoplamiento entre las matrices y obtener la matriz de chequeo de un código SC-LDPC se modifican las conexiones de modo que los nodos variables se conecten a nodos de

3. Análisis Teórico

chequeo de m_s *protographs* vecinos (figura 3.6, donde $m_s = 2$), manteniendo así el grado de todos los nodos.

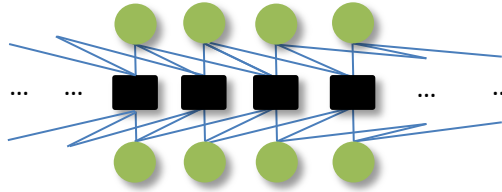


Figura 3.6: Cadena de *protographs* acoplados

Terminación del código

Para construir el código terminado con L *protographs* acoplados es necesario añadir m_s nodos de chequeo extra, para poder replicar los patrones de conexión. El código convolucional terminado (figura 3.7) puede ser tratado como un código LDPC-BC con palabras código de longitud $n=LM$ bits.

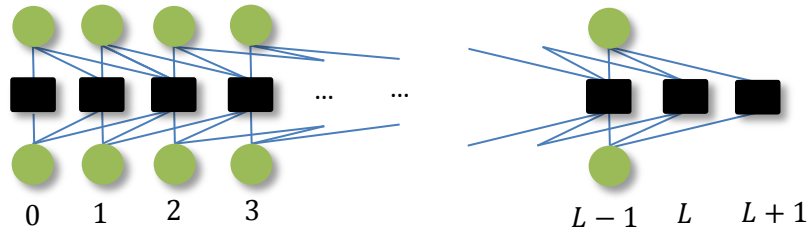


Figura 3.7: Estructura de un código SC-LDPC terminado

El hecho de terminar los códigos introduce una pequeña irregularidad estructurada en el grafo: los nodos de chequeo de los extremos tienen un grado menor, por lo que los mensajes enviados por éstos a sus vecinos son más fiables. Este efecto se propaga a lo largo del grafo progresivamente durante el proceso de decodificación iterativo, hacia los nodos situados en el centro.

3.2.3 Construcción de la matriz de chequeo de paridad

En esta sección se describe la *construcción clásica* de un código SC-LDPC regular (\mathbf{J} , \mathbf{K}) [23], utilizada para obtener las matrices de chequeo de los códigos.

3. Análisis Teórico

Sea a el máximo común divisor entre J, K :

$$a = m.c.d(J, K) \quad (3.27)$$

Existen los números enteros J', K' que cumplen:

$$m.c.d(J', K') = 1, \quad J = aJ', K = aK' \quad (3.28)$$

La tasa del SC-LDPC no terminado es:

$$R = 1 - \frac{J'}{K'} \quad (3.29)$$

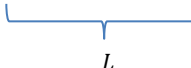
Al terminarse el código se sufre una pérdida de tasa que se hace inapreciable en la medida en la que $L \rightarrow \infty$:

$$R_L = 1 - \frac{L + m_s}{L} (1 - R) \quad (3.30)$$

$$\lim_{L \rightarrow \infty} R_L = R \quad (3.31)$$

Siendo $m_s = a - 1$ la memoria del código SC-LDPC y asumiendo que el código convolucional se termina tras L instantes de tiempo – hay un total de L *protographs* acoplados –, se define una matriz base \mathbf{B}_L que representa el código:

$$B_L = \begin{pmatrix} B_0 & & & \\ B_1 & B_0 & & \\ & \ddots & & \\ \dots & B_1 & & B_0 \\ B_{m_s} & \vdots & \ddots & B_1 \\ & B_{m_s} & \ddots & \vdots \\ & & & B_{m_s} \end{pmatrix} \quad (3.32)$$



3. Análisis Teórico

Las submatrices B_i que componen la matriz base \mathbf{B}_L tienen dimensiones $[J' \times K']$, y todos sus elementos son iguales a '1'. En consecuencia, las dimensiones totales de la matriz \mathbf{B}_L son $[J' \times (L + m_s)] \times [K' \times L]$.

El número de bits en cada posición es $M_d = MK'/J'$. Cada elemento de la submatriz \mathbf{B}_L se hace corresponder con un bloque de tamaño $[M_d] \times [M_d]$ de la matriz de chequeo: si el elemento (i, j) de la matriz \mathbf{B}_L es no nulo, el bloque de la matriz \mathbf{H} asociado es una permutación aleatoria de la matriz identidad $\mathbf{I}_{M_d \times M_d}$. En caso de ser nulo, el bloque asociado es un bloque de ceros.

La matriz de chequeo generada tiene unas dimensiones $[M_d \times J' \times (L + m_s)] \times [M_d \times K' \times L] = [M_d \times J' \times (L + m_s)] \times [M \times J' \times L]$. En la *figura 3.8* se muestra la estructura de la matriz de un código SC-LDPC (3,6) con $M = 1000$ y $L=20$, construida utilizando el método descrito:

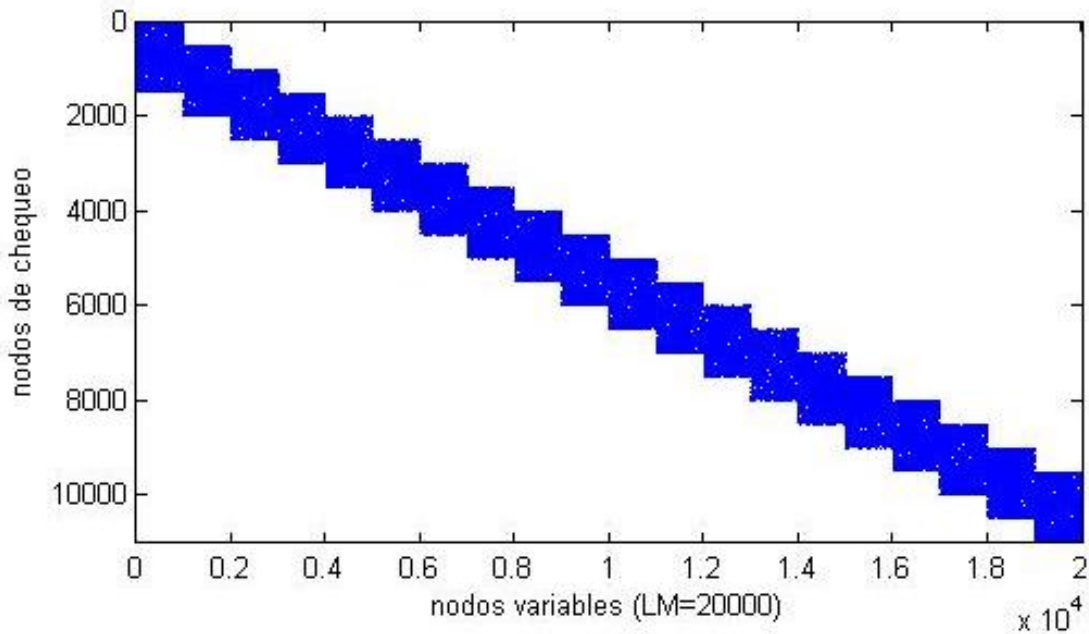


Figura 3.8: Matriz H de un SC-LDPC (3,6), $M=1000$, $L=20$

En la *figura 3.9* se representa el grado de cada uno de los nodos de chequeo en el código (3,6) anterior, apreciándose la irregularidad estructurada en los extremos:

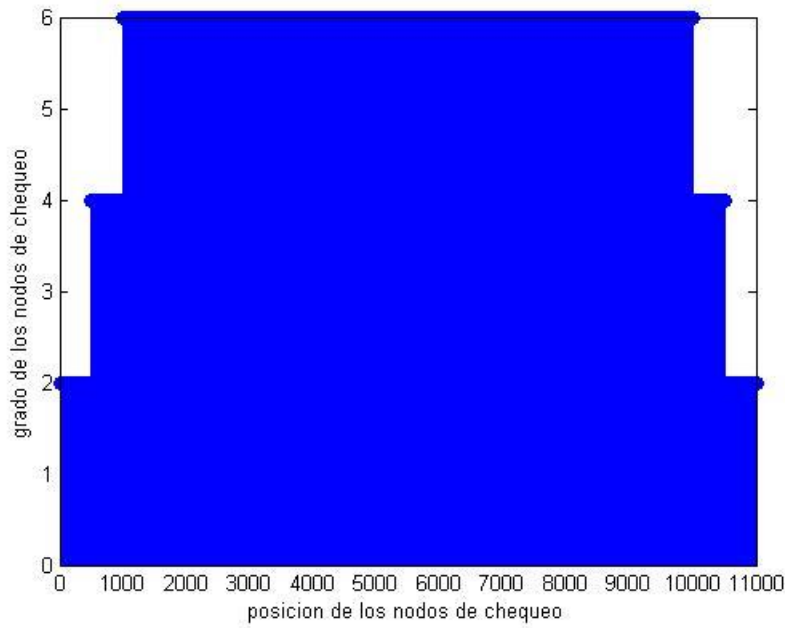


Figura 3.9: Grado de los nodos de chequeo en el código de la figura 3.8

3.2.4 Decodificación en ventanas

En el caso de los códigos SC-LDPC terminados se pueden decodificar los símbolos recibidos de modo similar a un código bloque: el decodificador bloque puede comenzar a realizar el proceso cuando se recibe la palabra código completa. El tiempo total de decodificación en este caso es:

$$T = T_{\text{dec}} + T_c \quad (3.33)$$

siendo T_c el tiempo que se tarda en recibir una palabra código y T_{dec} el tiempo de decodificación del bloque completo.

Las prestaciones de estos códigos mejoran a medida que aumenta la longitud de la palabra código. Para muchas aplicaciones en las que la latencia es un factor crítico – entre ellas las comunicaciones ópticas –, el retardo puede ser demasiado elevado. Otra cuestión a tener en cuenta es que la decodificación BP en códigos demasiado largos acarrea una complejidad computacional elevada. Para solventar estos problemas se plantea utilizar **decodificación iterativa enventanada**. Este esquema de decodificación aprovecha la estructura de la matriz de

3. Análisis Teórico

chequeo y permite alcanzar una solución de compromiso entre prestaciones, complejidad y latencia [24].

La estructura diagonal en bloques de los códigos SC-LDPC impone una restricción sobre los nodos variables que pueden estar conectados a un determinado nodo de chequeo: dos nodos variables distanciados más de $(m_s + 1) * J' * M$ posiciones no intervienen en una misma ecuación de comprobación. Esta característica puede utilizarse para realizar una decodificación continua sobre el flujo de símbolos recibidos utilizando una “ventana” que se desplaza sobre la secuencia: el decodificador opera aplicando el algoritmo BP en las ventanas, de modo semejante a un decodificador bloque.

El tamaño de ventana W determina el número de conjuntos de $J' * M$ nodos variables que se incluyen: cada ventana contiene $W * J' * M$ nodos variables y $W * J' * M_d$ nodos de chequeo. Además de los $W * J' * M$ nodos variables asociados a la ventana es necesario tener en cuenta también los $m_s * J' * M$ nodos anteriores, ya que en esta región los nodos variables también intervienen en las ecuaciones de chequeo en cuestión.

En la *figura 3.10* se especifica la submatriz de \mathbf{H} de un código SC-LDPC(3,6) con $M = 1000$ ($M_d = 500$), $L = 20$ y $m_s = 2$ con la que trabajaría el decodificador para una ventana $W=4$, en el quinto instante de decodificación:

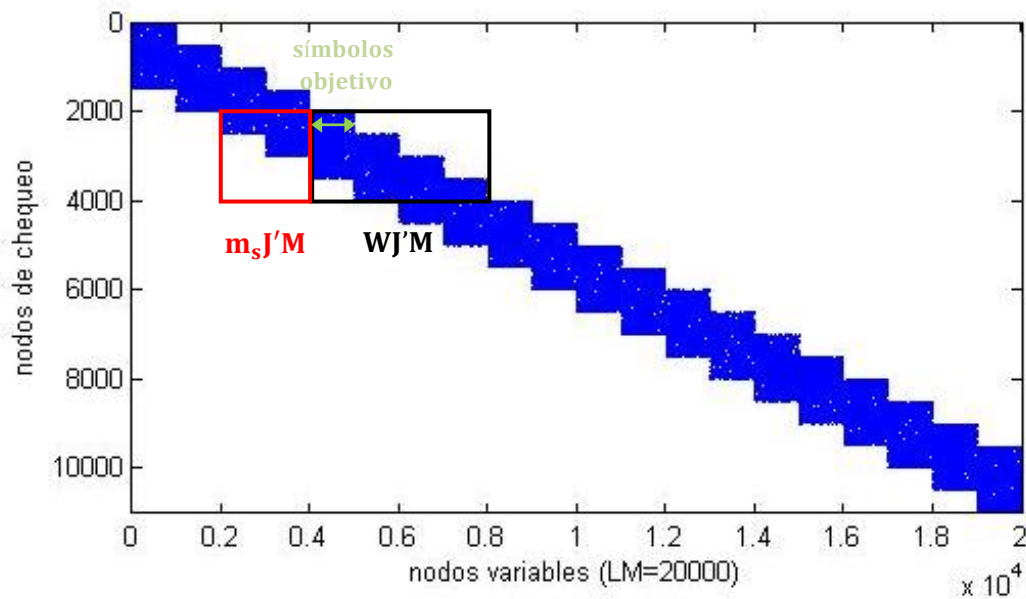


Figura 3.10: Ventana de decodificación en quinta iteración, $W=4$, $m_s=2$

3. Análisis Teórico

En cada posición de la ventana se decodifican los primeros $J'M$ símbolos o *símbolos objetivo*. El retardo de decodificación asociado a cada una de las ventanas es:

$$T = T_{\text{dec}}(W) + T_w \quad (3.34)$$

donde T_w es el tiempo que se tarda en recibir los símbolos asociados a la venta y T_{dec} el tiempo que se tarda en decodificarlos. Al utilizar el algoritmo BP sobre matrices de menores dimensiones que la matriz de chequeo de paridad completa se consigue reducir el tiempo de decodificación.

Una vez se han decodificado los símbolos objetivo, la ventana se desplaza $J' * M$ posiciones hacia la derecha. La *figura 3.11* muestra la ventana de decodificación del ejemplo anterior, en el sexto instante de decodificación:

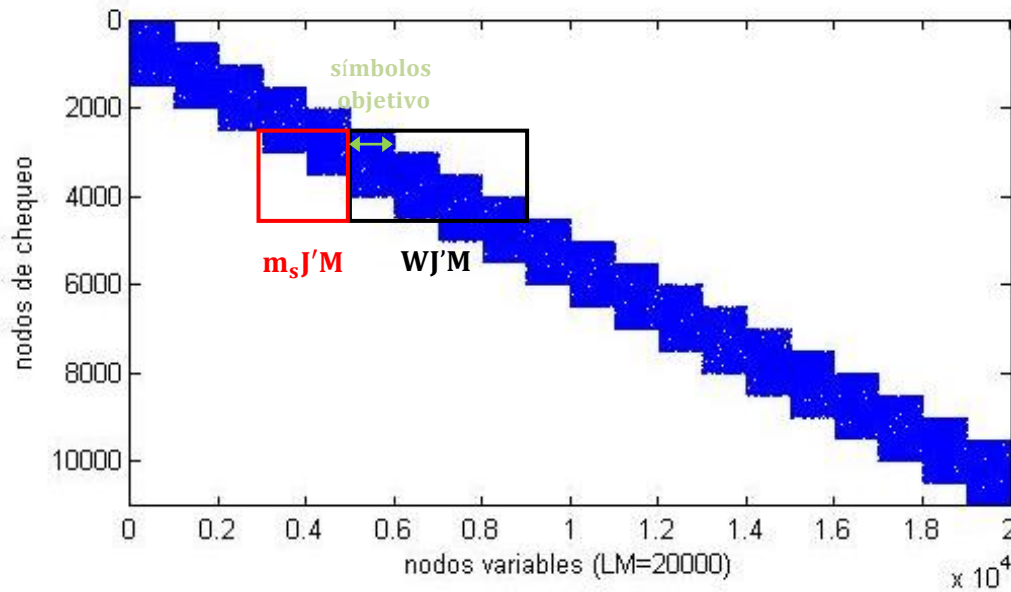


Figura 3.11: Ventana de decodificación en sexta iteración, $W=4$, $ms=2$

Se identifican varios parámetros de diseño asociados a la decodificación en ventanas:

-Tamaño de ventana W: es posible flexibilizar la latencia variando el parámetro W. Puede reducirse el retardo de decodificación disminuyendo la ventana a costa de empeorarse las prestaciones en términos de la probabilidad de error. Con el tamaño de ventana ha de alcanzarse una solución de compromiso entre retardo y prestaciones. Sería posible también implementar algoritmos de enventanado adaptativos que incrementen el tamaño de ventana si después de un

3. Análisis Teórico

número máximo de iteraciones del algoritmo BP no ha sido posible decodificar los *símbolos objetivo* o en los que el tamaño de ventana en cada desplazamiento dependa de los resultados obtenidos en las anteriores.

-Estrategia de propagación de información al desplazar la ventana: en las ecuaciones de chequeo de cada ventana intervienen también los $m_s J' M$ símbolos anteriores a los *símbolos objetivo*, decodificados en instantes previos. En este trabajo se comparan dos estrategias para utilizar esta información en cada instante de decodificación: **propagación dura** (*hard*), con la que se propagan los valores de estos bits después de decidir si es '1' o '0', y **propagación blanda** (*soft*), en la que se propagan las estadísticas LLR calculadas en la ventana anterior.

-Criterio de detención del algoritmo BP en cada desplazamiento de ventana: es necesario establecer un criterio con el que se decida cuándo se han decodificado correctamente los *símbolos objetivo* y ha de desplazarse la ventana. Pueden utilizarse diversos criterios de parada: desplazar la ventana cuando se alcancen un número fijo de iteraciones con el algoritmo BP, utilizar una regla basada en las estadísticas LLR de los *símbolos objetivo* (establecer un umbral de decodificación y realizar el desplazamiento si un determinado porcentaje de símbolos alcanza el umbral, o cuando se alcancen un número máximo de iteraciones con el BP) o utilizar una regla similar a la que utilizan los códigos LDPC-BC sobre los símbolos objetivo. En el proyecto se realiza un número fijo de iteraciones en cada ventana.

4

Análisis de prestaciones de códigos SC-LDPC

Contenidos

4.1 Entorno de simulación.....	50
4.1.1 Entorno de simulación Matlab/MEX	
4.1.2 Modelado del sistema de comunicaciones: simplificaciones	
4.1.3 Arquitectura del emulador	
4.1.4 Evaluación de las prestaciones	
4.2 Análisis de códigos SC-LDPC con decodificación en ventanas.....	56
4.2.1 Propagación dura SC-LDPC(3,24)	
4.2.2 Propagación blanda SC-LDPC(3,24)	
4.2.3 Códigos SC-LDPC con diferentes tasas. Comparativa con QC-LDPC	

En este capítulo se evalúan las prestaciones de los códigos SC-LDPC con decodificación en ventanas, con el objetivo de compararlos con técnicas de codificación de canal de primera y segunda generación y con otras propuestas de tercera generación.

En el primer apartado se exponen las características principales y arquitectura del entorno de simulación que modela el sistema de comunicaciones, desarrollado utilizando Matlab y lenguaje C. En la segunda sección se realiza un análisis de las prestaciones obtenidas con los códigos SC-LDPC y las variaciones en las mismas al modificar diferentes parámetros de los códigos: el factor de ensanchado, el tamaño de ventana, el número de iteraciones del algoritmo BP, la estrategia de propagación de información en el desplazamiento de la ventana y los grados de los nodos de chequeo y paridad. Se concluye con una comparación con los códigos propuestos en [15] y [19].

4.1 ENTORNO DE SIMULACIÓN

Se implementa un emulador software para estimar las prestaciones de los códigos SC-LDPC mediante el método de simulación de Monte Carlo. En este punto se describen las consideraciones que ha sido necesario realizar para que los resultados puedan ser obtenidos en un tiempo razonable, posteriormente se tratan las simplificaciones tenidas en cuenta para modelar el sistema de comunicaciones y la arquitectura del emulador, y finalmente qué parámetros son utilizados para evaluar dichas prestaciones.

4.1.1 Entorno de simulación Matlab/MEX

El factor limitante en la elaboración de este tipo de aplicaciones es el tiempo requerido en simulación: para que sea viable obtener resultados con códigos de largas longitudes es necesario desarrollarlas tratando de aumentar la velocidad de simulación y reducir la complejidad computacional y los requisitos de memoria en la medida de lo posible.

Matlab y MEX files

Se utiliza Matlab como entorno para desarrollar la plataforma de simulación del sistema de comunicaciones y analizar las prestaciones en términos de probabilidad de error de bit. Matlab no es óptimo para realizar determinados cálculos que se requieren en el proceso de

decodificación: el uso de numerosos bucles encadenados es demasiado lento, lo que motiva utilizar programas escritos en otro lenguaje para disminuir el tiempo total de ejecución.

En este caso en concreto es útil que Matlab interaccione con programas externos para acelerar las simulaciones, a través del uso de ficheros MEX (*Matlab EXecutable*): programas o funciones compiladas escritos en C, C++ o FORTRAN que pueden integrarse en el sistema y ser ejecutados desde Matlab como si se tratase de una función propia. Se han utilizado ficheros MEX para establecer una interfaz con el lenguaje de programación C y aligerar el proceso de decodificación, tanto para realizar el desplazamiento de la ventana como en la aplicación del algoritmo iterativo BP en cada una de ellas.

No sólo se implementan las rutinas computacionales que ejecutan las instrucciones a realizar; también es necesario implementar una rutina de enlace que funcione como interfaz cuando la función MEX es llamada desde Matlab, para transferir los parámetros necesarios. El intercambio se realiza a través de cuatro variables: dos matrices que contienen los parámetros de entrada y salida, y el número de argumentos de entrada y salida. En esta función se verifican el número de argumentos de entrada y salida y su coherencia, se obtienen los argumentos de entrada, se reserva espacio para las matrices que contienen los argumentos de salida y se llama a la rutina computacional escrita en C.

Reducción de complejidad computacional

El principal cuello de botella del proceso de decodificación es la búsqueda de los nodos vecinos a los que se conectan cada uno de los nodos variables y de chequeo, y a los que envían mensajes en cada iteración del algoritmo. No resulta viable realizar todo el proceso sobre la matriz de chequeo de paridad \mathbf{H} completa con códigos de grandes longitudes, por lo que se simplifica la carga computacional trabajando en su lugar con dos matrices que contienen las posiciones de los nodos vecinos: una de ellas con tantas columnas como nodos variables hay y con un número de filas igual a su grado en la que se almacenan las posiciones de los nodos de chequeo vecinos, y otra con tantas filas como nodos de chequeo existen y con un número de columnas igual a su grado que contiene la ubicación de los nodos variables vecinos.

Uso de memoria

El hecho de utilizar matrices de menores dimensiones que la matriz \mathbf{H} reduce los requisitos de memoria del sistema: la decodificación se realiza utilizando las dos matrices de posiciones de vecinos, y modificando los límites de las submatrices sobre las que opera el decodificador a medida que se desplaza la ventana. También se ha tenido en cuenta que las operaciones de reserva y liberación de memoria, necesaria para almacenar el contenido de los mensajes, tienen

un gran coste. Para limitar este impacto sólo se realizan en una ocasión cada vez que desde Matlab se llama a la función de decodificación, y se reutiliza la memoria ya reservada a medida que se desplaza la ventana.

4.1.2 Modelado del sistema de comunicaciones: simplificaciones

El sistema de comunicaciones modelado es simplificado. Se considera una modulación binaria BPSK (*Binary Phase-Shift Keying*) en lugar de una modulación DP-QPSK, como se emplea en [14] para recomendar la máxima redundancia que puede ser añadida con el esquema de protección y corrección de errores. A pesar de no ser una modulación utilizada en los escenarios de transmisión ópticos es un esquema extendido en el análisis de prestaciones de técnicas FEC y es semejante al utilizado en el artículo [19], tomado como referencia para contrastar el funcionamiento de los códigos SC-LDPC con el de los códigos de primera y segunda generación y con los códigos QC-LDPC de tercera generación presentados en el mismo.

Por otro lado se simplifica el modelado del canal óptico asumiendo un canal gaussiano. El ruido tras la detección óptica no posee una estadística gaussiana: las principales fuentes de ruido son la emisión espontánea amplificada, la distorsión de los pulsos debida a las no linealidades de la fibra, la dispersión cromática, la interferencia entre símbolos... [25]. No obstante muchos de los estudios de técnicas FEC en sistemas de transmisión de fibra óptica utilizan una aproximación de ruido gaussiano y modelan el canal óptico como un canal gaussiano para analizar el funcionamiento de los códigos en la región de SNR de interés [19].

4.1.3 Arquitectura del emulador

El emulador *software* consta de dos bloques: el **generador de código** y el **simulador de prestaciones del código**.

Generador de código

La matriz de chequeo de paridad se genera a partir de un procedimiento sencillo y clásico basado en *protographs* expuesto en el *apartado 3.2.3* del proyecto. Se construye la matriz de chequeo a partir de los siguientes parámetros: el factor de expansión **M**, el número de *protographs* acoplados **L** y los grados de los nodos variables y de chequeo (**J, K**).

4. Modelado práctico

Posteriormente se construyen las matrices de posiciones de vecinos y éstas son almacenadas en un fichero junto con el resto de parámetros del código.

Simulador de prestaciones del código

Se implementa el esquema de simulación mostrado en la *figura 4.1*:

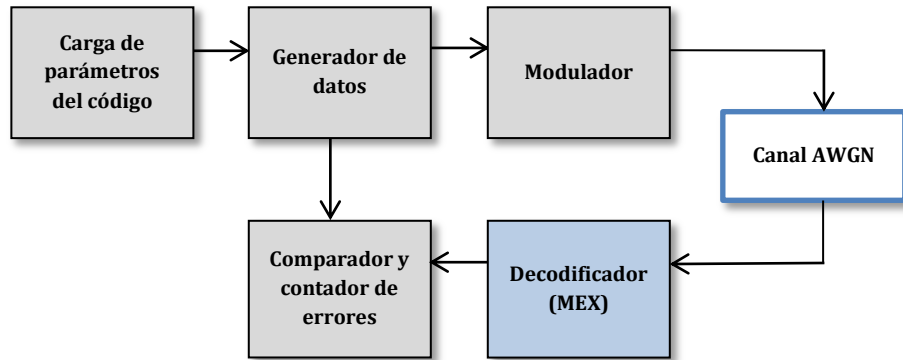


Figura 4.1: Diagrama de bloques del sistema de comunicaciones modelado con Matlab/MEX

Carga de parámetros del código: se cargan los parámetros y matrices de posiciones de vecinos de un código generado previamente.

Generador de datos a transmitir: se genera un paquete de datos binarios a transmitir a través del canal. Para simplificar el esquema y evitar calcular la matriz generadora \mathbf{G} y realizar la etapa de codificación se envía siempre la palabra código con todos los bits iguales a '0'.

Modulador: se utiliza una modulación BPSK. Los bits '0' y '1' se representan realizando el mapeo '0' \rightarrow '-1', '1' \rightarrow '+1' para hacer la transmisión a través del canal.

Canal: se asume ruido aditivo, blanco gaussiano (*Additive White Gaussian Noise* AWGN) con media nula y varianza σ^2 .

Decodificador: el decodificador, implementado en C e incorporado al sistema a través de los ficheros MEX, recibe la señal tras atravesar el canal. El simulador permite realizar decodificación bloque o decodificación en ventanas utilizando el algoritmo de decodificación blando *Belief Propagation* presentado en el capítulo tercero. Los parámetros de entrada son: los **parámetros del código** previamente cargado, el **número de iteraciones** que realiza el algoritmo BP y el **tamaño de ventana** en caso de que la

decodificación sea enventanada. Si la decodificación se realiza en ventanas, es posible propagar la información asociada a los símbolos anteriores a los *símbolos objetivo* que intervienen en las ecuaciones de chequeo utilizando dos estrategias: **propagación dura** (decisiones de dichos símbolos efectuadas en ventanas anteriores) o **propagación blanda** (estimaciones del logaritmo de cociente de verosimilitudes asociadas a cada símbolo, obtenidas en ventanas previas).

En el paso de inicialización del algoritmo BP los nodos variables son inicializados con el término intrínseco de la *ecuación 3.20*. Con el modelo de transmisión binaria y ruido gaussiano empleado, las funciones de densidad de probabilidad condicionadas son en cada caso:

$$p(r_n | c_n = 1) = p(r_n | t_n = 1) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(r_n - 1)^2}{2\sigma^2}\right) \quad (4.1)$$

$$p(r_n | c_n = 0) = p(r_n | t_n = -1) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(r_n + 1)^2}{2\sigma^2}\right) \quad (4.2)$$

El término de inicialización, asumiendo bits equiprobables, es por tanto:

$$\begin{aligned} \log \frac{p(r_n | c_n = 1)}{p(r_n | c_n = 0)} &= \log \frac{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(r_n - 1)^2}{2\sigma^2}\right)}{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(r_n + 1)^2}{2\sigma^2}\right)} = \\ &= \log \left(\exp \left(-\frac{(r_n - 1)^2}{2\sigma^2} + \frac{(r_n + 1)^2}{2\sigma^2} \right) \right) = \frac{2r_n}{\sigma^2} \end{aligned} \quad (4.3)$$

No se ha establecido una condición que al satisfacerse detenga el proceso de decodificación: tras un número fijo de iteraciones, se decide $c_n = 1$ si el cociente de verosimilitudes estimado $LLR_{c_n} = \lambda_n^{[l]} > 0$ y $c_n = 0$ en caso contrario.

Comparador y contador de errores: el número de palabras código a transmitir y decodificar es otro parámetro de simulación. Se calcula la probabilidad de error de bit comparando la palabra enviada con la palabra decodificada, y contando posteriormente el número de errores.

4.1.4 Evaluación de las prestaciones

Típicamente se utilizan la tasa del código R y la relación entre un bit de mensaje y ruido E_bN_0 para analizar las prestaciones de los códigos de protección contra errores. En este trabajo se lleva a cabo dicho análisis con la relación E_bN_0 . No obstante, en el ámbito de las comunicaciones ópticas es frecuente utilizar una figura de mérito Q en su lugar:

$$Q = E_bN_0 + 10 \log_{10} 2R \quad (4.4)$$

Se tiene en cuenta también que los códigos con tasas de codificación menores proporcionan mejores prestaciones al tener mayor redundancia, por lo que no es justo comparar códigos con diferentes tasas para las mismas probabilidades de error en el canal. Al realizar una codificación con una tasa R , se incrementa la energía efectiva que se utiliza para transmitir cada bit. Por tanto para obtener las curvas BER con cada código se utiliza la relación señal a ruido $ebn0'$ después de codificar:

$$ebn0 = 10^{\frac{EbN0}{10}} \quad (4.5)$$

$$ebn0' = ebn0 * R \quad (4.6)$$

En todas las gráficas en el eje de abscisas se representa la relación E_bN_0 sin codificar, y en el eje de ordenadas la probabilidad de error de bit. Se incluye también como referencia la curva de probabilidad de error del canal sin codificar para comprobar cómo están funcionando los códigos:

$$P_e = \frac{1}{2} \operatorname{erfc}(\sqrt{ebn0}) \quad (4.7)$$

4.2 ANÁLISIS DE CÓDIGOS SC-LDPC CON DECODIFICACIÓN EN VENTANAS

Se analizan las prestaciones de diversos códigos SC-LDPC, eligiendo tasas adecuadas para comunicaciones ópticas de alta velocidad. Los códigos de primera generación tienen una redundancia del 7%, pero a medida que la velocidad de la electrónica se incrementa es posible implementar esquemas FEC más potentes, con mayor redundancia: el OIF recomienda un límite del 20%.

En este apartado se estiman y comparan las prestaciones de códigos **SC-LDPC(3,15)**, **SC-LDPC(3,24)**, **SC-LDPC(4,20)** y **SC-LDPC(4,32)**. En la *tabla 4.1* se resumen las tasas de los códigos propuestos, en función de los grados de los nodos de chequeo y variables y el número de *protographs* acoplados:

Código SC-LDPC (J,K)	Número de <i>protographs</i> acoplados L	Memoria del código ms	Tasa del código R	Redundancia (%)
(3,24)	48	2	0.8698	13.02%
(4,32)	47	3	0.8670	13.3%
(3,15)	48	2	0.7917	20.83%
(4,20)	47	3	0.7872	21.28%

Tabla 4.1: Tasas de los códigos SC-LDPC propuestos

En la *tabla 4.2* se presenta la longitud de las palabras código en función de los factores de ensanchado y el número de *protographs* acoplados utilizados en simulación en los siguientes subapartados, y en la *tabla 4.3* la longitud de las palabras con las que trabaja el decodificador si se realiza decodificación en ventanas, con los tamaños de ventana empleados:

Factor de ensanchado M	Número de <i>protographs</i> acoplados L	Longitud de palabra código (decodificación bloque)
640	47	30080
	48	30720
960	47	45120
	48	46080
1280	47	60160
	48	61440

Tabla 4.2: Longitudes de las palabras código en función de M y L

4. Modelado práctico

Factor de ensanchado M	Tamaño de ventana W	Longitud de palabra asociada a la ventana
640	10	6400
	15	9600
	20	12800
	30	19200
960	10	9600
	15	14400
	20	19200
	30	28800
1280	10	12800
	15	19200
	20	25600
	30	38400

Tabla 4.3: Longitudes de las palabras a decodificar en función de M y W

En el Apéndice C se incluyen los tiempos de decodificación de una palabra código en simulación, para cada uno de los códigos propuestos, con los factores de ensanchado, tamaños de ventana y número de iteraciones utilizados en este apartado.

En primer lugar se toma como referencia un código SC-LDPC(3,24) para comparar las prestaciones obtenidas al realizar propagación dura y propagación blanda en los desplazamientos de ventana. A continuación se utiliza ese mismo código para analizar diversos aspectos con propagación blanda: se verifica que se produce el efecto de decodificación desde los extremos hacia el centro tanto con el decodificador bloque como con el decodificador en ventanas, y se analiza el efecto que tiene la variación del número de iteraciones, el factor de ensanchado y el número de ventanas.

A partir de los resultados obtenidos se eligen el número de iteraciones y tamaños de ventana que serán utilizados en el apartado 4.2.3 para comparar las prestaciones de los códigos SC-LDPC con diferentes grados con los propuestos en [15] y [19].

4.2.1 Propagación dura SC-LDPC(3,24)

En este punto se realiza una comparativa entre la utilización de **propagación de decisiones duras** y **propagación blanda** de la información asociada a los símbolos decodificados en ventanas anteriores utilizando un código SC-LDPC(3,24) con $L=48$ *protographs* acoplados. En ambos casos se realiza **decodificación blanda** con el algoritmo BP. En la *figura 4.2* se representa la BER con propagación blanda y propagación dura, realizando 20 iteraciones en el algoritmo BP en cada una de las ventanas:

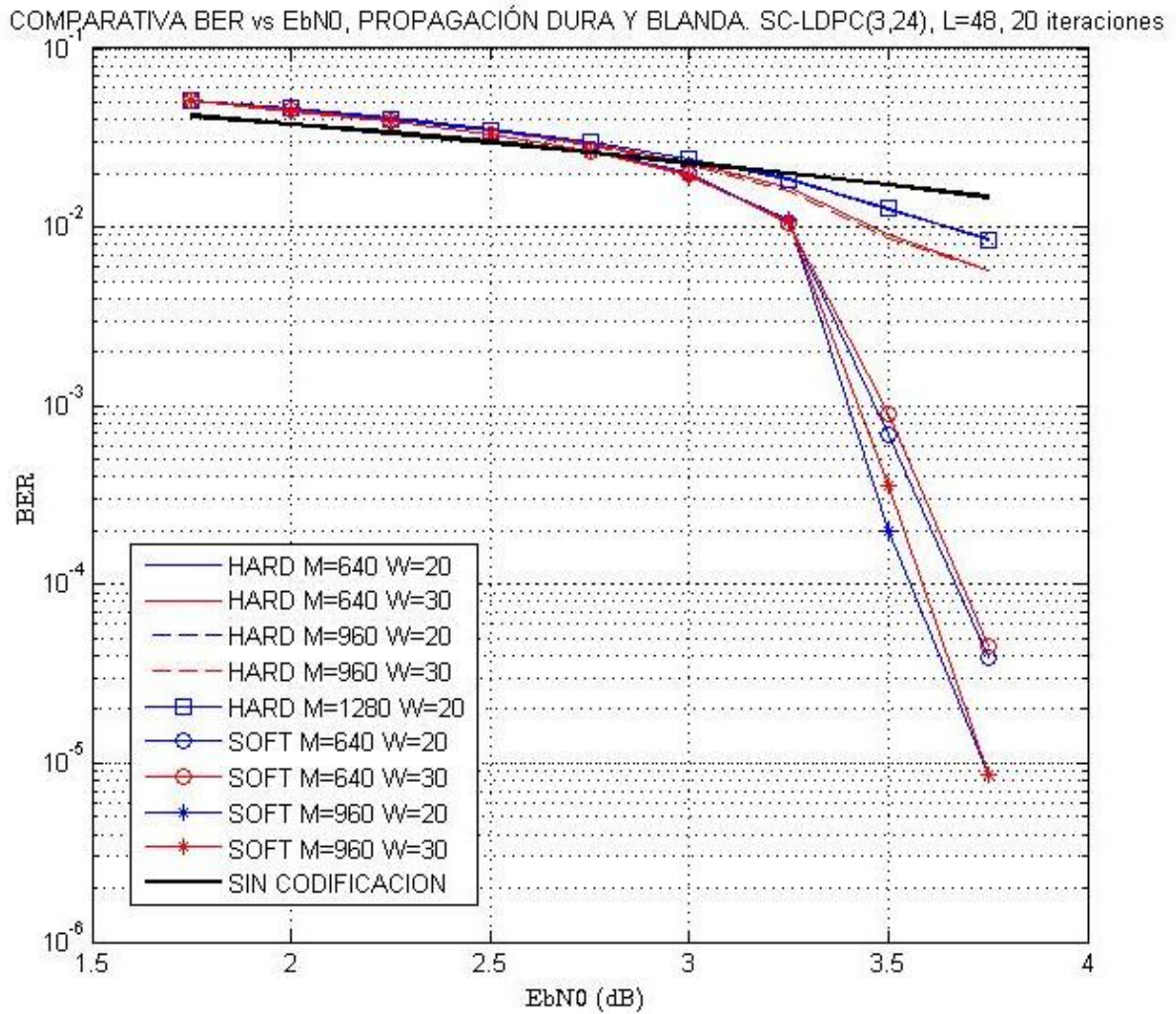


Figura 4.2: Comparativa propagación soft/hard, para SC-LDPC(3,24), L=48, 20 iteraciones

4. Modelado práctico

En la *figura 4.3* se realiza una comparativa para el caso de propagación dura entre códigos con diferentes factores de ensanchado y utilizando en decodificación tamaños de ventana $W=20$ y $W=30$, con 20 y 40 iteraciones en el algoritmo BP.

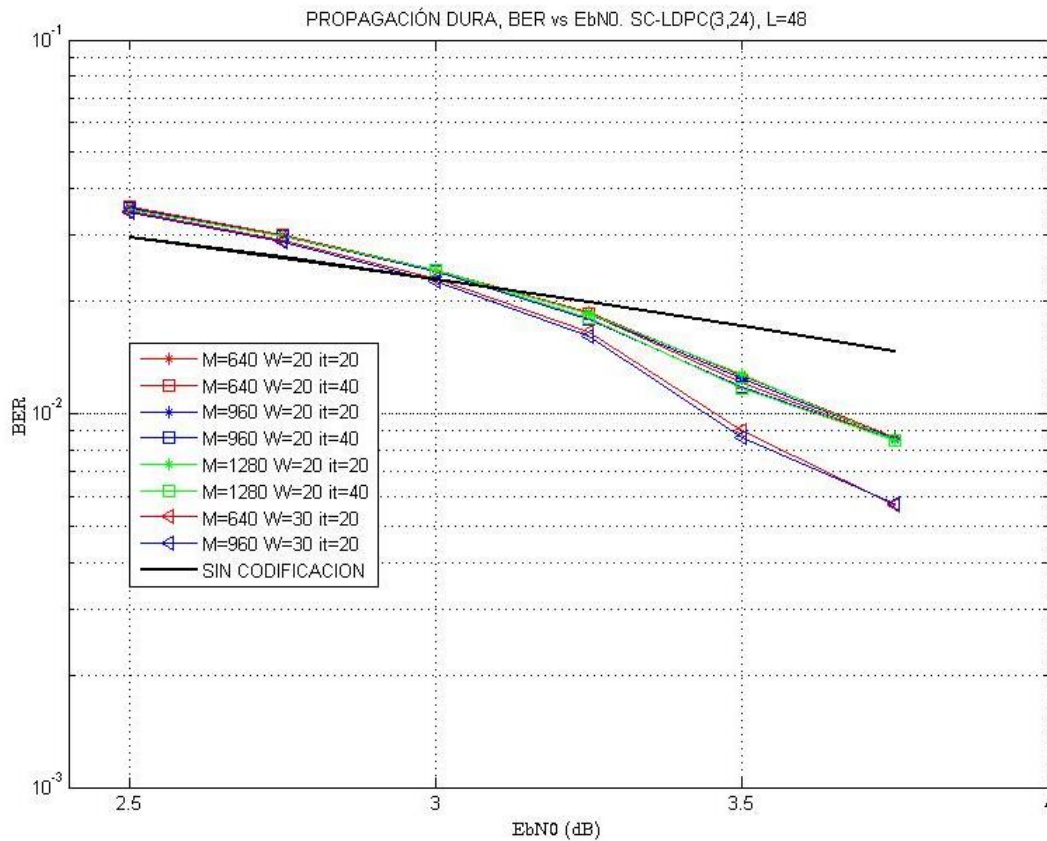


Figura 4.3: Comparativa propagación hard, para SC-LDPC(3,24), L=48

Se percibe una ligera mejora al aumentar el número de iteraciones y el factor de ensanchado para un tamaño de ventana fijo. Las prestaciones mejoran al incrementar el tamaño de ventana, pero son mucho peores a las obtenidas con propagación blanda.

4.2.2 Propagación blanda SC-LDPC(3,24)

4.2.2.1 Efecto de decodificación desde los extremos hacia el centro

Se verifica que se produce el efecto de decodificación desde los extremos hacia el centro, debido a las irregularidades estructuradas en los grados de los nodos de chequeo introducidas al terminarse el código tras L instantes de tiempo.

En la *figura 4.4* se representa la LLR asociada a cada uno de los bits de la palabra código (a partir de la cual se decide si cada uno de los bits recibidos es '1' o '0'), estimada tras realizar un número determinado de iteraciones variable con el algoritmo BP y un decodificador bloque. Las gráficas son obtenidas con un código SC-LDPC(3,24), un factor de ensanchado $M=640$ y un número de *protographs* $L=20$, para una $E_b/N_0=3$ dB.

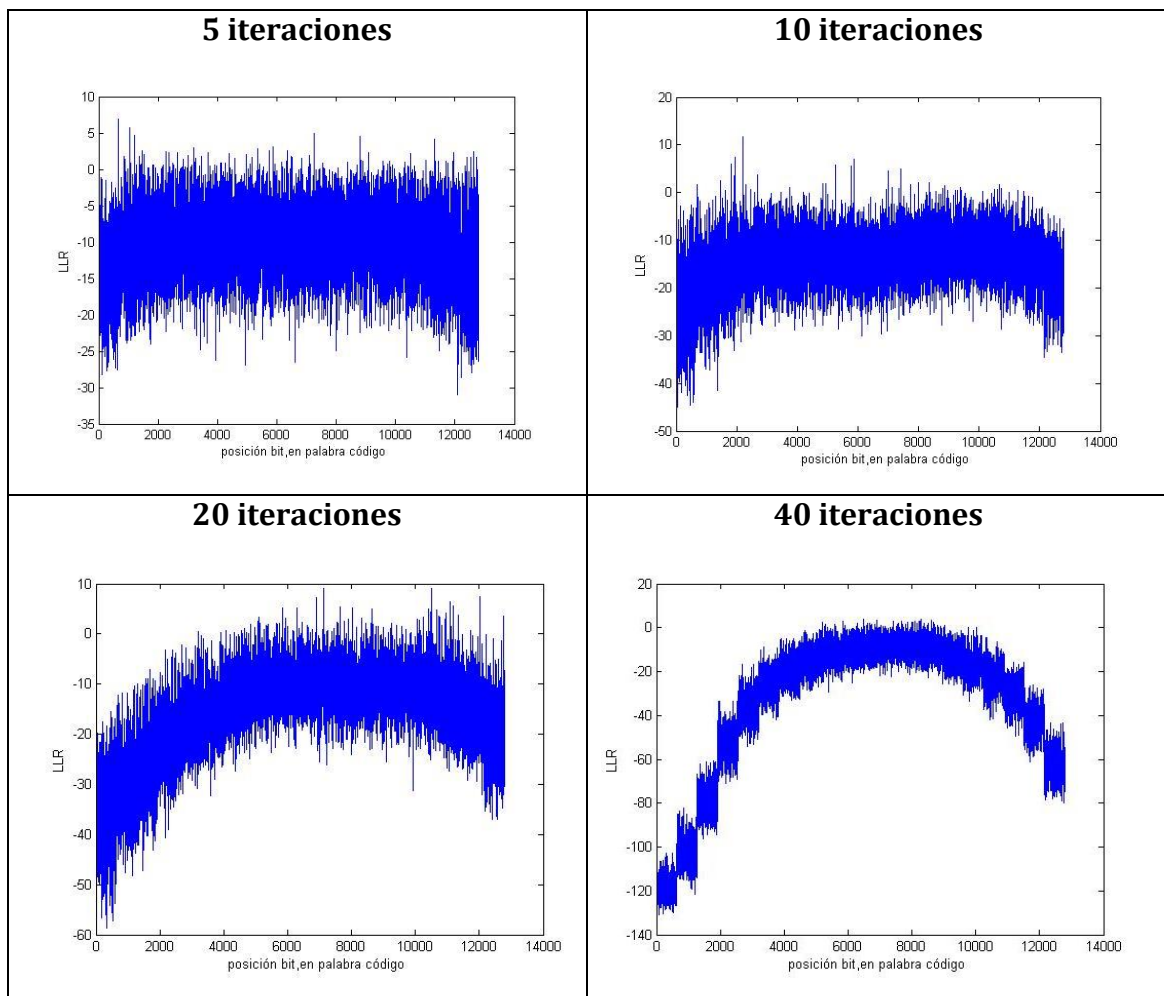


Figura 4.4: LLR tras un número variable de iteraciones, con decodificación bloque

4. Modelado práctico

Los nodos de chequeo de los extremos tienen grados menores, de modo que la información enviada por éstos a sus vecinos es más fiable. Se observa cómo el efecto se propaga desde los nodos de los extremos hacia los nodos del centro a medida que se incrementa el número de iteraciones.

En la *figura 4.5* se muestra que este efecto se mantiene al realizar decodificación en ventanas con propagación blanda. Se utiliza un código SC-LDPC(3,24) con $M=640$, $L=48$ y $W=20$ para una $E_b/N_0=3$ dB, de modo que la longitud de las palabras en cada ventana es igual a la longitud de la palabra código en el caso de decodificación bloque del ejemplo anterior.

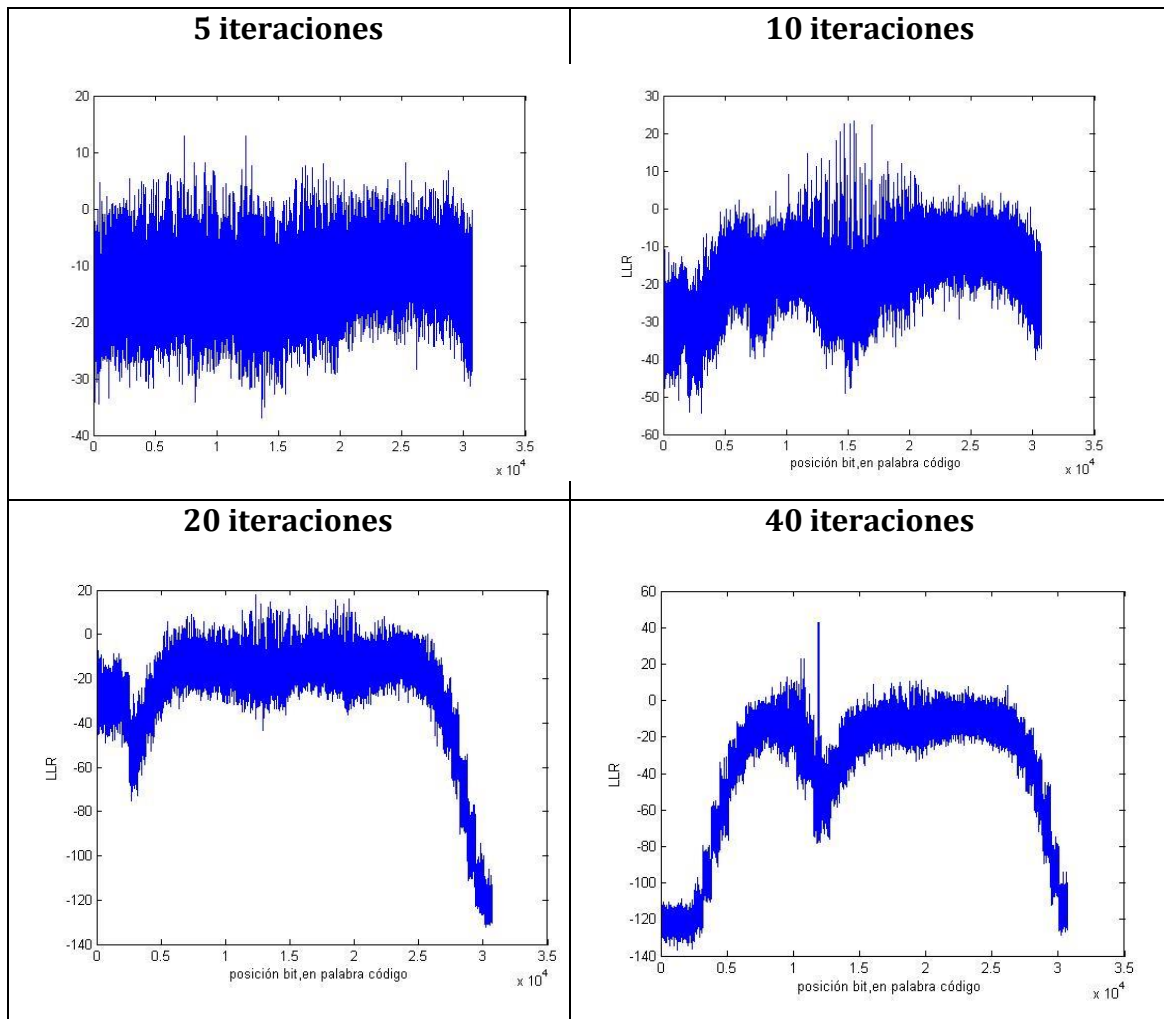


Figura 4.5: LLR tras un número variable de iteraciones, con decodificación en ventanas

4. Modelado práctico

4.2.2.2 Número de iteraciones

Se compara la BER de códigos SC-LDPC(3,24) con factores de ensanchado $M=640$, $M=960$ y $M=1280$, realizando decodificación en ventanas con un tamaño de ventana fijo y con 20 y 40 iteraciones en el algoritmo BP. En la *figura 4.6* se representan las curvas BER obtenidas en simulación:

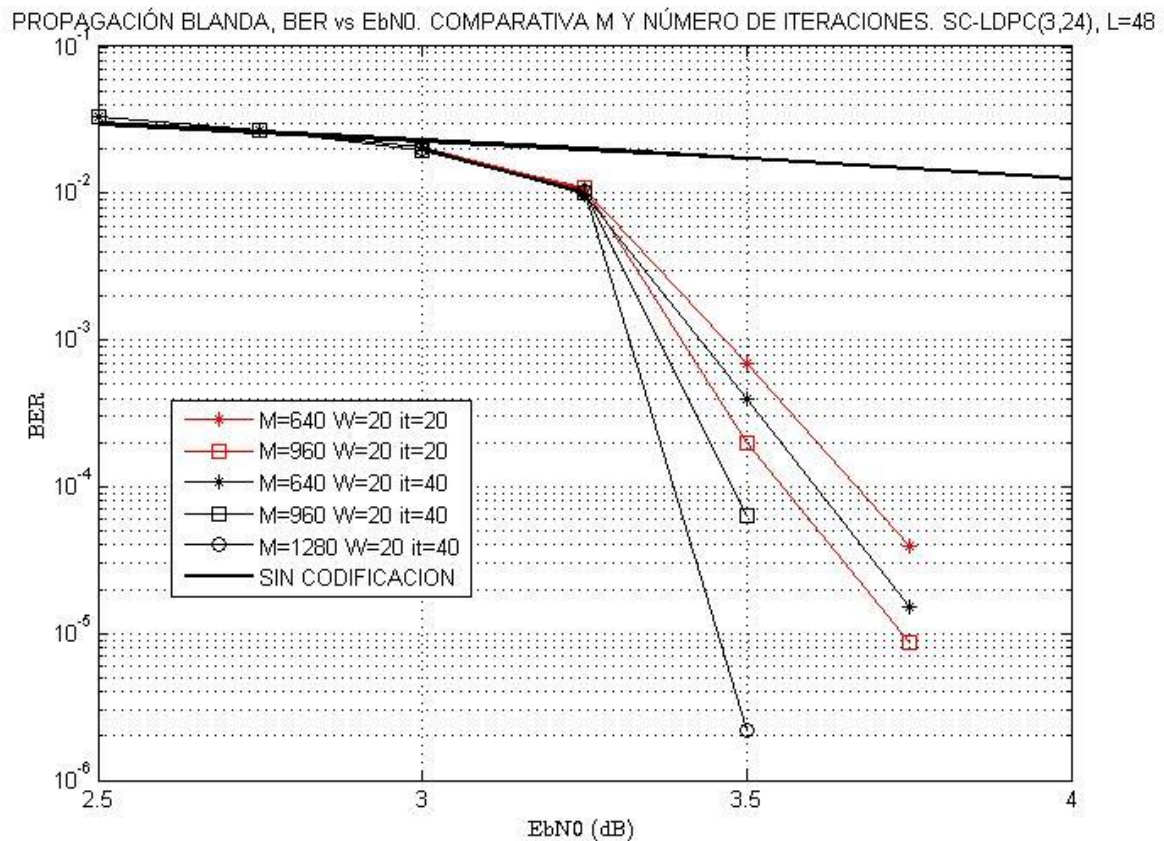


Figura 4.6: Comparativa propagación soft, variación it, para SC-LDPC(3,24), L=48.

En la gráfica se puede comprobar que a medida que se incrementa la longitud de la palabra código las curvas caen con una pendiente más pronunciada. Se aprecia que realizar 20 iteraciones en cada una de las ventanas no es suficiente: las prestaciones mejoran considerablemente al realizar 40 iteraciones. No se han obtenido diferencias notables al seguir aumentando el número de iteraciones, por lo que se ha decidido utilizar 40 iteraciones para realizar la comparación en el apartado 4.2.3.

4. Modelado práctico

4.2.2.3 Tamaño de ventana

A continuación en la *figura 4.5* se compara la BER de códigos SC-LDPC(3,24) con factores de ensanchado $M=640$, $M=960$, tamaños de ventana $W=20, W=30$ y 20 iteraciones en cada ventana. En la *figura 4.7* se incluyen las curvas BER simuladas:

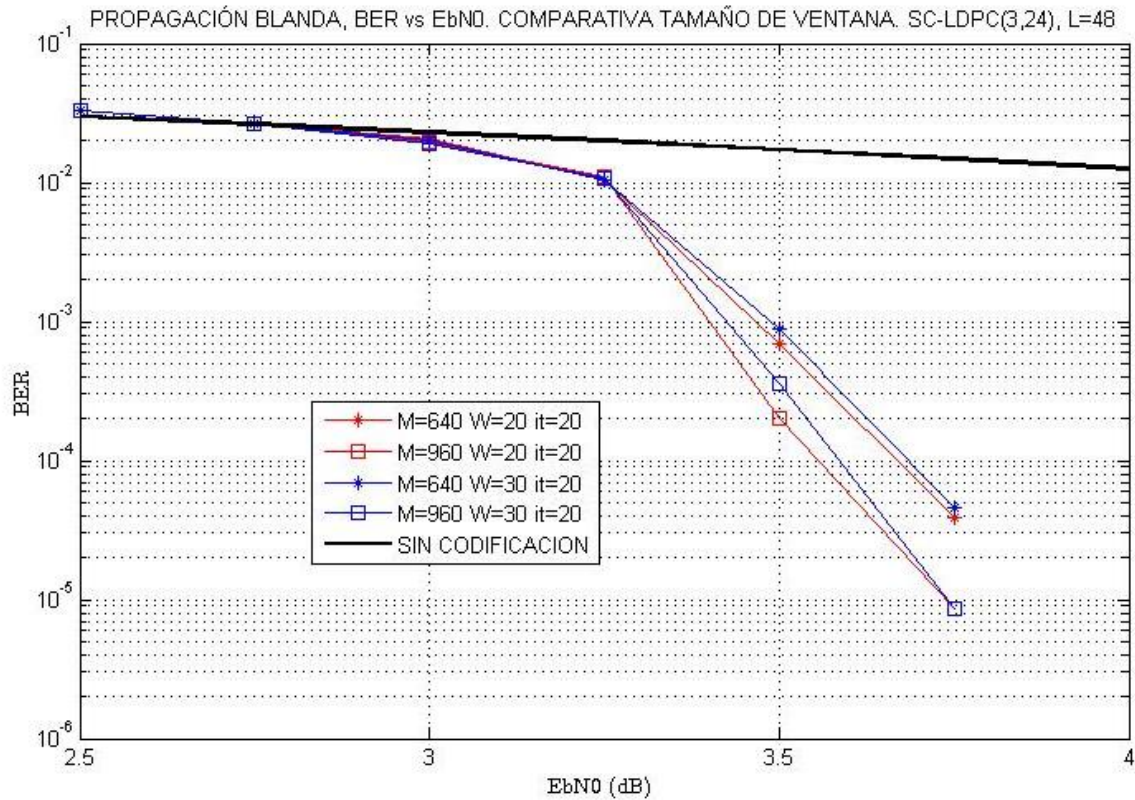


Figura 4.7: Comparativa propagación soft, variación W , para SC-LDPC(3,24), $L=48$, 20 iteraciones

Cabe esperar que al incrementar el tamaño de ventana las curvas caigan con una pendiente mayor ya que la longitud de la palabra en cada ventana es superior. Este efecto se encuentra enmascarado en la figura superior, al utilizarse un número de iteraciones insuficiente.

4. Modelado práctico

En la figura 4.8 se representan las probabilidades de error de bit del mismo código, realizando 40 iteraciones en cada una de las ventanas, con tamaños de ventana $W=10$, $W=15$ y $W=20$.

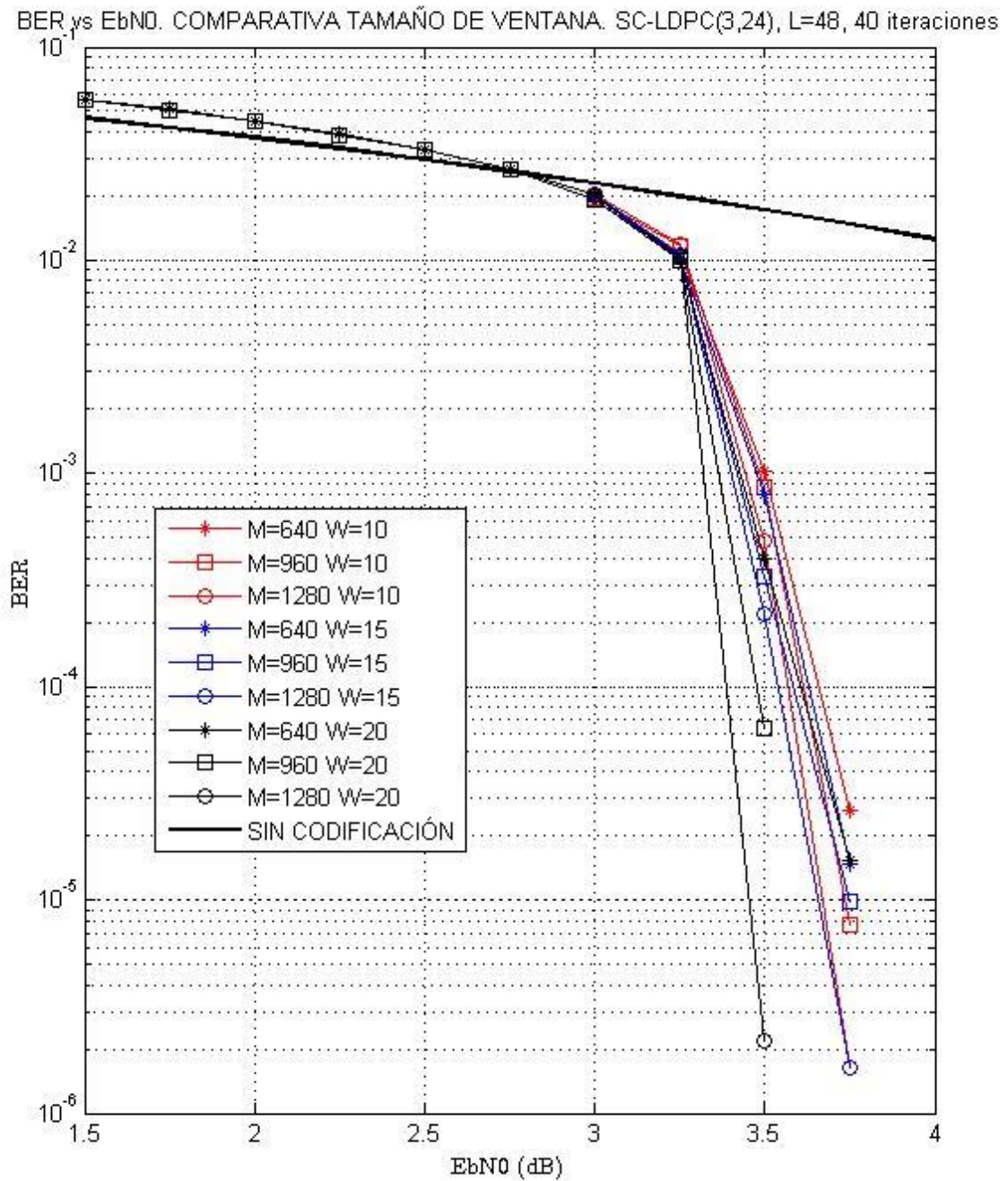


Figura 4.8: Comparativa propagación soft, variación W , para SC-LDPC(3,24), $L=48$, 40 iteraciones

Las gráficas corroboran que para un tamaño de ventana fijo las prestaciones mejoran a medida que se incrementa el factor de ensanchado M , y lo mismo ocurre al aumentar el tamaño de ventana para un determinado valor de M .

4. Modelado práctico

Se ha decidido fijar un tamaño de ventana $W=20$ para hacer la comparativa inicial entre los códigos con diferentes grados en el apartado 4.2.3, de modo que si es posible incrementar la latencia de decodificación y utilizar palabras con un número más elevado de bits se incrementa el número de bits por posición (M) en lugar del número de bloques que se decodifican en cada ventana (W).

4.2.3 Códigos SC-LDPC con diferentes tasas. Comparativa con QC-LDPC

Se comparan códigos con diferentes tasas seleccionando diferentes distribuciones de los grados: SC-LDPC(3,15), SC-LDPC(3,24), SC-LDPC(4,20) y SC-LDPC(4,32). Se realizan 40 iteraciones en cada una de las ventanas, y se utiliza un tamaño de ventana $W=20$. Se simula un número variable de bits por posición $M=640$, $M=960$ y $M=1280$. En la figura 4.9 se muestran los resultados obtenidos:

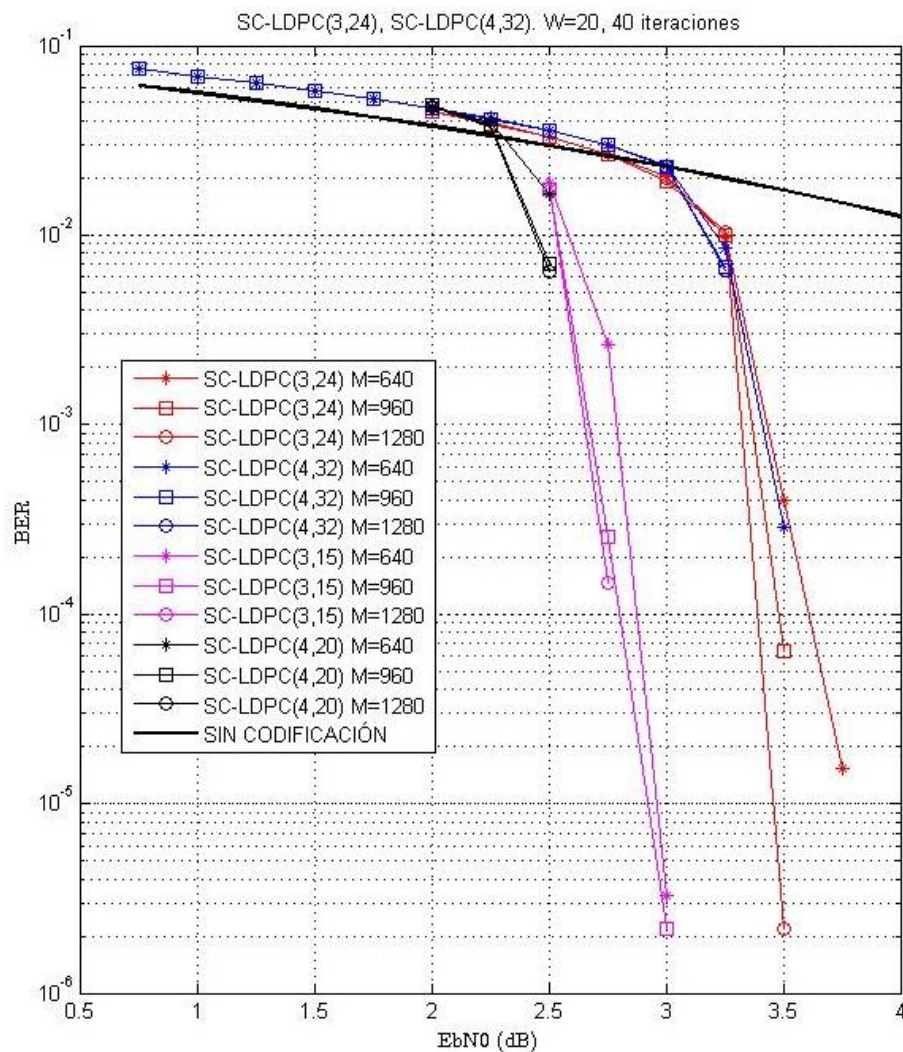


Figura 4.9: Comparativa SC-LDPC con diferentes grados

4. Modelado práctico

Se obtienen mayores ganancias de codificación con los códigos SC-LDPC (3,15) y SC-LDPC(4,20) que con los códigos SC-LDPC(3,24) y SC-LDPC(4,32), ya que las tasas de codificación de los dos primeros son menores (se introduce más redundancia). También se observa que las prestaciones en términos de la probabilidad de error mejoran al utilizar nodos con mayores grados para códigos con tasas similares, a cambio de una mayor complejidad en decodificación: las ganancias obtenidas son superiores con el SC-LDPC(4,20) que con el SC-LDPC(3,15), y del mismo modo son mejores en el caso del SC-LDPC(4,32) respecto al SC-LDPC(3,24).

En el caso del código SC-LDPC(4,32) con $M=1280$ no se ha obtenido ningún error al simular 20000 palabras código para una $E_bN_0 = 3.5$ dB, lo que implica que la BER en este punto es inferior a $\frac{1}{M \cdot L \cdot 20000} = \frac{1}{1280 \cdot 47 \cdot 20000} = 8.3112 \cdot 10^{-10}$. En vista de la baja probabilidad de error obtenida en este caso se evalúan también las prestaciones con tamaños de ventana $W=10$ y $W=15$, representadas en la *figura 4.10*:

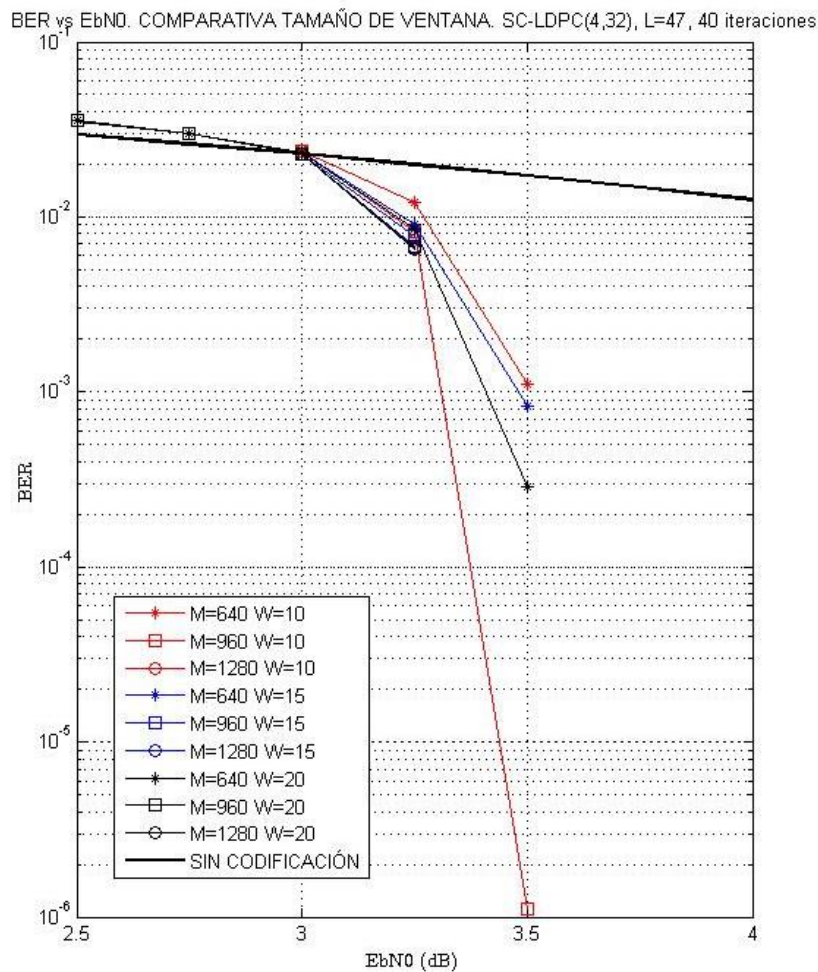


Figura 4.10: Comparativa SC-LDPC (4,32)

4. Modelado práctico

Finalmente se realiza la comparación con los códigos QC-LDPC propuestos en [19]: dos códigos QC-LDPC irregulares, uno de ellos con tasa 0.9375 (6.25% de redundancia) y longitud de palabra código 32640 bits y el otro con una tasa 0.833 (redundancia de 16.7%) y una longitud de la palabra código de 12240. En el código de tasa 0.9375 los nodos de chequeo tienen grado 62 y los nodos variables grado 2 o 4, y en el caso del código con tasa 0.833 los nodos de chequeo tienen grado 22 y los variables 2 o 4. Se incluye en la comparación también la curva de un código QC-LDPC [15] con palabra código de 18356 bits y una tasa 0.833 (16.7% de redundancia), la primera propuesta para sistemas ópticos de 100 Gb/s para la que se verificó mediante un emulador basado Arrays de Compuertas Lógicas Programables (*Field Programmable Gate Array*, FPGA) que no se produce *error floor* para BER superiores a 10^{-15} .

En la *figura 4.11* se añaden las curvas de los siguientes códigos SC-LDPC:

- SC-LDPC(3,24) con tasa de codificación $R=0.8698$ (13.02% de redundancia), $M=1280$ y $W=20$, en el que el decodificador en ventanas trabaja con palabras de 25600 bits.
- SC-LDPC(4,32) con tasa $R=0.8670$ (13.3% de redundancia), $M=960$ y $W=10$, en el que el decodificador opera con palabras de 9600 bits.
- SC-LDPC(4,32) con tasa de codificación $R=0.8670$ (13.3% de redundancia), $M=1280$ y $W=20$, donde el decodificador trabaja con palabras de 25600 bits. Al no haberse obtenido ningún error en simulación para $E_bN_0 = 3.5$ dB transmitiendo 20000 palabras, se añade en línea discontinua la mínima pendiente que tendrá la curva BER (la probabilidad de error estimada sería como máximo $8.3112 * 10^{-10}$ en dicho punto).

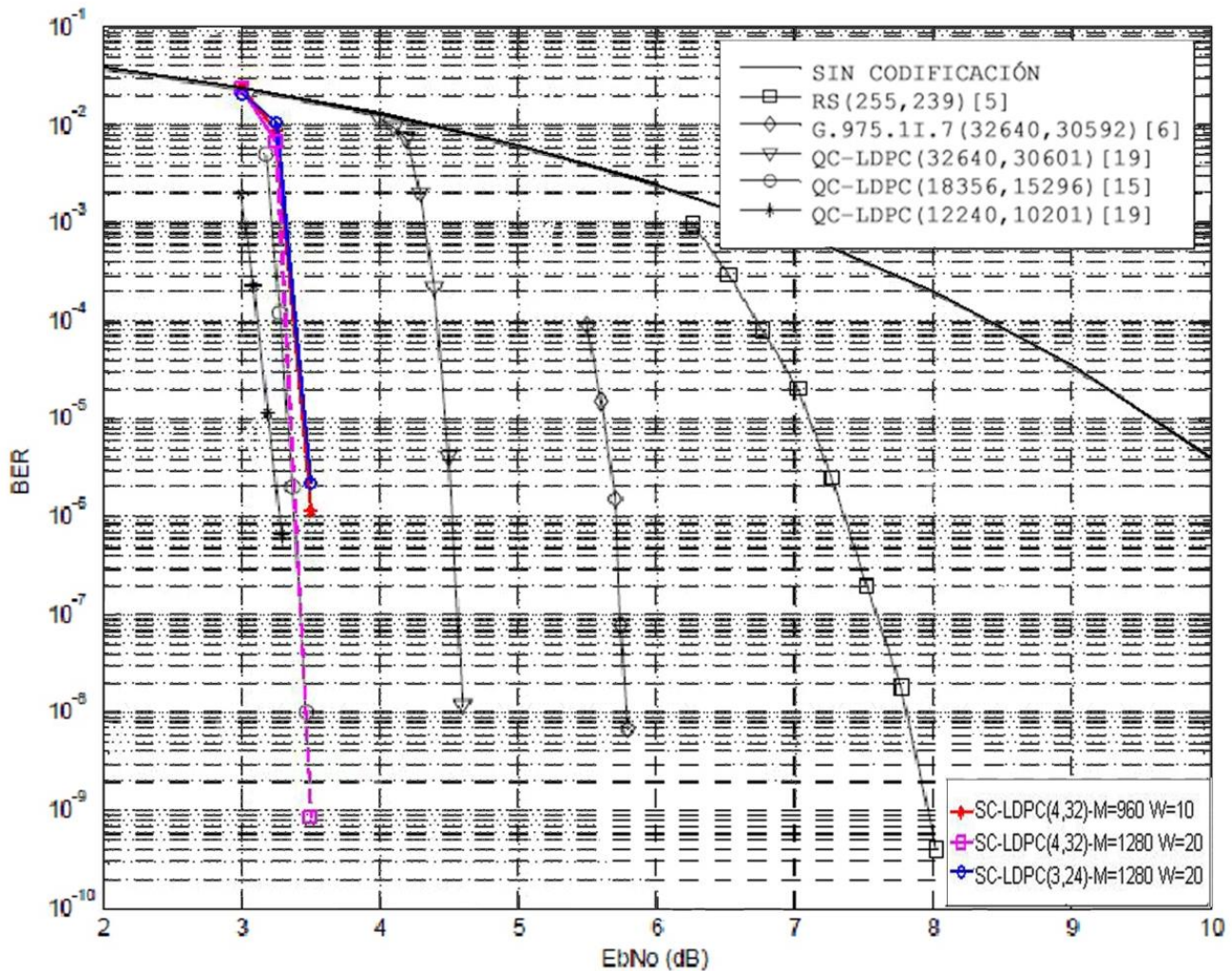


Figura 4.11: Comparativa SC-LDPC con QC-LDPC[19] y [15], y códigos de primera y segunda generación

En la gráfica se aprecia que con los códigos SC-LDPC se obtienen unas presentaciones mejores a las ofrecidas por los códigos de primera y segunda generación. Además, son competitivos con los códigos QC-LDPC de tercera generación con los que se realiza la comparación en la región de E_b/N_0 analizada. En concreto con el SC-LDPC(4,32) con $M=1280$ y $W=20$ se mejoran las prestaciones del código QC-LDPC presentado en [15] introduciendo una menor redundancia.

Se debe resaltar que la construcción de los códigos es lo más sencilla posible: únicamente se tienen en cuenta las distribuciones de los grados de los nodos. No se ha llevado a cabo ningún procedimiento de diseño u optimización ni se ha efectuado un análisis en las conexiones de los grafos, lo que mejoraría las prestaciones ofrecidas. Se verifica que es viable considerar los

4. Modelado práctico

códigos SC-LDPC como candidatos a ser introducidos en futuros estándares pero todavía quedan muchos aspectos por resolver, entre ellos estudiar un procedimiento de construcción de la matriz de chequeo óptimo, evaluar las prestaciones de los códigos en la región de *error floor* o analizar en profundidad las ventajas que ofrece la estructura de la matriz de chequeo de estos códigos a la hora de realizar una implementación práctica del codificador y decodificador.

5

Conclusiones y líneas futuras

Contenidos

5.1 Conclusiones.....	72
5.2 Líneas de trabajo futuras.....	74
5.2.1 Simulación software	
5.2.2 Simulación hardware	

A continuación se presenta un análisis de los principales resultados obtenidos en este trabajo y se proponen posibles líneas de investigación vinculadas al mismo.

5.1 CONCLUSIONES

El uso de un código corrector de errores en recepción es esencial en los sistemas de comunicaciones de fibra óptica modernos, que requieren velocidades de transmisión cada vez más elevadas. Actualmente se proponen múltiples técnicas para ser introducidas en las redes ópticas de transporte de próximas generaciones, en su mayoría códigos basados en grafos y técnicas que utilizan decisión blanda. En este trabajo se analizan los códigos convolucionales LDPC con Acoplamiento Espacial, ya que poseen características que los hacen apropiados para los sistemas de comunicaciones ópticas: proporcionan buenas prestaciones en la región *waterfall* y de *error floor*, la estructura *quasi*-regular y localizada de los “unos” en la matriz de chequeo de paridad facilita una arquitectura *hardware* del decodificador eficiente, el algoritmo de decodificación iterativo posee una baja complejidad, y se puede flexibilizar la latencia de decodificación si se utiliza un decodificador en ventanas.

En particular, se analizan las prestaciones de los códigos SC-LDPC utilizando una representación basada en *protographs* para describir su construcción y un decodificador en ventanas basado en un algoritmo iterativo de paso de mensajes: el algoritmo *Belief Propagation*. Para ello se ha desarrollado un entorno de simulación *software* basado en Matlab/MEX con el que resulta viable en términos de tiempo de simulación calcular las curvas de probabilidad de error de bit en función de la relación energía por bit a ruido mediante el método de Monte Carlo, para códigos con palabras de decenas de miles de bits.

Siguiendo las recomendaciones del OIF, que propone como límite utilizar esquemas que añadan un 20% de redundancia, se estudia el comportamiento de códigos con diferentes grados y tasas próximas a 0.8, la tasa mínima recomendada: SC-LDPC(3,24), SC-LDPC(4,32), SC-LDPC(3,15) y SC-LDPC(4,20), con tasas 0.8698, 0.8670, 0.7917 y 0.7872, respectivamente.

5. Conclusiones y líneas futuras

Se han extraído las siguientes conclusiones mediante simulación, con el código SC-LDPC(3,24):

- Las prestaciones obtenidas realizando propagación dura a la hora de desplazar la ventana son mucho peores a las obtenidas llevando a cabo propagación blanda, utilizando en ambos casos decodificación blanda.

- Se ha verificado el efecto de decodificación desde los extremos hacia el centro tanto en el decodificador bloque como en el decodificador en ventanas, causado por la irregularidad estructurada que existe en los grados de los nodos de chequeo en los extremos de la matriz de chequeo de paridad.

- A partir de los resultados obtenidos con este código se ha seleccionado un tamaño de ventana $W=20$ y 40 iteraciones buscando un compromiso entre retardo de decodificación y prestaciones, para realizar una comparativa con otras técnicas FEC propuestas hasta la fecha.

Finalmente se comparan las prestaciones ofrecidas por los códigos SC-LDPC y los códigos presentados en [15] y [19]. Se ha comprobado que los códigos SC-LDPC mejoran las prestaciones de los códigos de primera y segunda generación y son competitivos con los códigos QC-LDPC de tercera generación propuestos en los artículos.

Los resultados obtenidos muestran que los códigos SC-LDPC son competitivos en las regiones de E_b/N_0 analizadas y que son candidatos a ser introducidos en próximos estándares de comunicaciones ópticas. Cabe destacar que la construcción de los códigos utilizada es sencilla y únicamente se tienen en cuenta las distribuciones de los grados de los nodos, no se ha llevado a cabo ningún procedimiento de optimización ni análisis en las conexiones de los grafos para evitar ciclos de longitudes cortas, lo que mejoraría las prestaciones en la región *waterfall*, ni otras estructuras que pueden originar *error floor* al realizarse decodificación iterativa.

5.2 LÍNEAS DE TRABAJO FUTURO

Se proponen diversas cuestiones no resueltas en el proyecto que han de estudiarse antes de cumplir el objetivo final: diseñar un código que proporcione las prestaciones adecuadas para las aplicaciones de comunicaciones ópticas, y unas arquitecturas del codificador y decodificador en las que se preste atención a la eficiencia (área y consumo de potencia). Se dividen los temas todavía abiertos en dos bloques: simulación *software* y *hardware*.

5.2.1 Simulación *software*

Se plantea estudiar las siguientes cuestiones, realizando variaciones sobre el emulador *software* implementado:

- Realizar una **comparativa entre la complejidad y las prestaciones** de los códigos **LDPC-BC** y **SC-LDPC**, para determinar si los códigos SC-LDPC con decodificación en ventanas proporcionan mejores prestaciones que los códigos bloque en términos de complejidad y probabilidad de error, igualando el número de bits incluidos en la ventana a la longitud de la palabra del código bloque y utilizando en ambos las mismas distribuciones de los grados de los nodos variables y de chequeo.
- Proponer una **regla de finalización anticipada** del algoritmo BP para determinar cuándo se han realizado suficientes iteraciones en una ventana, buscando una solución de compromiso entre latencia y prestaciones: para relaciones señal a ruido elevadas para las que la palabra recibida contiene pocos errores, el proceso iterativo puede finalizar antes de que se efectúen todas las iteraciones. Por ejemplo, se plantea utilizar una regla basada en establecer un umbral en las estadísticas LLR de los *símbolos objetivo* o utilizar una regla similar a la empleada por los códigos LDPC-BC, basada en verificar que se cumplen las ecuaciones de chequeo de paridad en las que intervienen los *símbolos objetivo*.
- Estudiar **algoritmos de enventanado adaptativo** que permiten flexibilizar la latencia, aumentando el tamaño de ventana si tras un número de iteraciones no ha sido posible decodificar los *símbolos objetivo* o seleccionando el tamaño en función de los resultados obtenidos en la ventana anterior.

- Utilizar un modelo de sistema más próximo a los escenarios de comunicaciones ópticas, simulando una modulación DP-QPSK y evaluando las alternativas al modelo de canal gaussiano empleado.

5.2.2 Simulación *hardware*

El análisis del *error floor* es fundamental a la hora de seleccionar un código entre las múltiples propuestas que existen: las aplicaciones de comunicaciones ópticas requieren conocer el comportamiento de los códigos cuando la BER es muy baja, en torno a 10^{-12} y preferiblemente 10^{-15} . Evitar que aparezca *error floor* para BER superiores es una de las mayores dificultades que surgen a la hora de diseñar códigos para estas aplicaciones.

Con un entorno de simulación *software* se requieren semanas de simulación para caracterizar la BER en dicha región: típicamente se evalúa en un entorno *hardware*, utilizando un emulador de sistema y un decodificador basados en FPGA.

Con esta plataforma se conseguiría **acelerar el proceso de simulación**. Una vez desarrollado este entorno se propone estudiar un diseño de la matriz de chequeo óptimo analizando las conexiones en los grafos, mejorándose así las prestaciones tanto en la región *waterfall* como el *error floor*. En caso de que se produzca *error floor* a una BER superior a la requerida se plantea evaluar el uso de un código concatenado similar a los utilizados en los esquemas de segunda generación, por ejemplo, antes de desplazar la ventana y propagar la información relacionada a los símbolos ya decodificados que intervienen en las ecuaciones de chequeo de la siguiente ventana de decodificación.

Apéndice

Apéndice A

GLOSARIO DE ACRÓNIMOS Y NOMENCLATURA

A.1 Acrónimos

ARQ	<i>Automatic Repeat reQuest</i>
ATM	<i>Asynchronous Transfer Mode</i>
AWGN	<i>Additive White Gaussian Noise</i>
BCH	<i>Código Bose- Chaudhuri-Hocquenghem</i>
BER	<i>Bit Error Rate</i>
BP	<i>Belief Propagation</i>
BPSK	<i>Binary Phase Shift Keying</i>
CSOC	<i>Convolutional Self Orthogonal Code</i>
DP-QPSK	<i>Dual Polarization Quadrature Phase Shift Keying</i>
DSP	<i>Digital Signal Processor</i>
DWDM	<i>Dense Wavelength Division Multiplexing</i>
FEC	<i>Forward Error Correction</i>
FPGA	<i>Field Programmable Gate Array</i>
IP	<i>Internet Protocol</i>
LAN	<i>Local Area Network</i>
LDPC	<i>Low Density Parity Check</i>
LDPC-BC	<i>LDPC-Block Codes</i>
LDPC-CC	<i>LDPC-Convolutional Codes</i>
LLR	<i>Log-Likelihood Ratio</i>
LSI	<i>Large Scale Integration</i>
MAP	<i>Maximum A Posteriori</i>
MEX	<i>Matlab EXecutable</i>
ML	<i>Maximum Likelihood</i>
NCG	<i>Net Coding Gain</i>
ODU	<i>Optical channel Data Unit</i>
OIF	<i>Optical Internetworking Forum</i>
OSNR	<i>Optical Signal to Noise Ratio</i>
OTN	<i>Optical Transport Network</i>
QC-LDPC	<i>Quasi cyclic-LDPC</i>
RS	<i>Reed Solomon</i>
SDH	<i>Synchronous Digital Hierarchy</i>
SNR	<i>Signal to Noise Ratio</i>
STM	<i>Synchronous Transport Module</i>
UIT	<i>Union Internacional de Telecomunicaciones</i>
UIT-T	<i>sector de normalización de las Telecomunicaciones de la UIT</i>
WDM	<i>Wavelength Division Multiplexing</i>
WiMAX	<i>Worldwide Interoperability for Microwave Access</i>

A.2 Nomenclatura

Variables y símbolos

a	Máximo común divisor entre los grados de los nodos de chequeo y paridad $a = m.c.d.(J, K)$
b	Mensaje de entrada del codificador
\hat{b}	Mensaje de salida del decodificador
c	Palabra código de salida del codificador
\hat{c}	Palabra código de salida del decodificador
c_n	Símbolo n -ésimo de la palabra código de salida del codificador
\hat{c}_n	Símbolo n -ésimo de la palabra código de salida del decodificador
C	Capacidad de canal
d_h	Distancia de Hamming
d_{min}	Distancia mínima
EbN0	Relación energía por bit a ruido
J	Grado de los nodos variables
J'	Número que cumple la ecuación $J = aJ'$
k	Número de símbolos del mensaje de entrada del codificador, b
K	Grado de los nodos de chequeo
K'	Número que cumple la ecuación $K = aK'$
L	Número de instantes tras los que se termina el código (número de <i>protographs</i> acoplados)
m, m_s	Memoria del código convolucional $m_s = a - 1$
M	Factor de ensanchado del código SC-LDPC $n = ML$
M_d	Número de bits que representa cada conjunto de nodos variables en un <i>protograph</i> $M_d = \frac{M}{n_v} = MK'/J'$
n	Número de símbolos de la palabra código, c
n_c	Conjuntos de nodos de chequeo en un <i>protograph</i>
n_v	Conjuntos de nodos variables en un <i>protograph</i>
NC	Nodo de chequeo
NV	Nodo variable
r	Palabra de entrada al decodificador
r_{cod}	Número de símbolos de redundancia introducidos por el codificado $r_{cod} = n - k$
r_n	Símbolo n -ésimo de la palabra de entrada al decodificador
R	Tasa de código $R = k/n$
R_L	Tasa de código SC-LDPC terminado $R_L = 1 - \frac{L+m_s}{L}(1 - R)$
T	Tiempo de decodificación total. Decodificación bloque: $T = T_{dec} + T_c$. Decodificación enventanada: $T = T_{dec}(W) + T_w$
T_c	Tiempo que se tarda en recibir una palabra código
T_{dec}	Tiempo de decodificación de un bloque completo
$T_{dec}(W)$	Tiempo de decodificación de una ventana
T_w	Tiempo que se tarda en recibir los símbolos asociados a una ventana
v_s	Longitud de decodificación
W	Tamaño de ventana. Número de conjuntos de $J' * M$ nodos variables que intervienen
\mathcal{X}	Alfabeto al que pertenecen los símbolos de los mensajes y palabras código $\mathcal{X} = \{0,1\}$
z_m	m -ésima ecuación de comprobación $z_m = \sum_{n \in N_m} c_n$

$z_{m,n}$	m -ésima ecuación de comprobación excluyendo el n -ésimo nodo variable $z_{m,n} = \sum_{i \in N_{m,n}} c_i$
$\eta_{m,n}$	Mensaje enviado del nodo de comprobación m -ésimo al nodo variable n -ésimo
λ_n	Mensaje enviado por el nodo variable n -ésimo a los nodos de comprobación vecinos
σ^2	Varianza de ruido

Conjuntos

\mathcal{C}	Conjunto de palabras código $\mathcal{C} = \{c_0, c_1, \dots, c_{2^k-1}\}$
N_m	Conjunto de nodos variables que intervienen en una ecuación de comprobación (elementos no nulos de la m -ésima fila de H) $N_m = \{n: H_{m,n} = 1\}$
$N_{m,n}$	Conjunto de nodos variables que intervienen en una ecuación de comprobación, excepto el n -ésimo $N_{m,n} = N_m \setminus n$
M_n	Conjunto de ecuaciones de comprobación en las que interviene el nodo variable n -ésimo (elementos no nulos de la n -ésima columna de H) $M_n = \{m: H_{m,n} = 1\}$
$M_{n,m}$	Conjunto de ecuaciones de comprobación en las que interviene el nodo variable n -ésimo excepto la ecuación m -ésima $M_{n,m} = M_n \setminus m$
$\{r_i, i \neq n\}$	Conjunto de símbolos de la palabra de entrada al decodificador, excepto el n -ésimo

Matrices

B	Matriz base de un <i>protograph</i> . Contiene el número de conexiones entre los nodos variables y de chequeo
B_L	Matriz base de un código SC-LDPC terminado $[J' * (L + m_s)] \times [K' * L]$
G	Matriz generadora $[k] \times [n]$
H	Matriz de comprobación de paridad $[n - k] \times [n]$
H_{CC}	Matriz de comprobación de paridad de un código convolucional
H_L	Matriz de comprobación de paridad terminada después de L instantes
I_k	Matriz identidad $[k] \times [k]$

Funciones matemáticas

$\operatorname{argmax}_x(f(x))$	Valor del argumento de x que maximiza la función $f(x)$
$\log(x)$	Logaritmo neperiano
$\tanh(x)$	Tangente hiperbólica
$\tanh^{-1}(x)$	Tangente hiperbólica inversa

Apéndice B

GLOSARIO DE REGLAS MATEMÁTICAS

B.1 Regla de Bayes

Puede reescribirse la función de densidad condicionada de A dado B $p(a/b)$, a partir de las distribuciones *a priori* $p(a)$, $p(b)$ y la función de densidad condicionada de B dado A $p(b/a)$:

$$p(a/b) = \frac{p(b/a)p(a)}{p(b)} \quad (\text{B.1})$$

B.2 Regla de la tangente

Se define el logaritmo del cociente de verosimilitudes de la variable aleatoria binaria \mathbf{x} :

$$\lambda(x) = \log \frac{p(x=1)}{p(x=0)} \quad (\text{B.2})$$

A partir de la ecuación anterior, sabiendo que $p(x=1) + p(x=0) = 1$, podemos reescribir:

$$p(x=1) = \log \frac{e^{\lambda(x)}}{1 + e^{\lambda(x)}} \quad (\text{B.2})$$

Si \mathbf{x}_1 y \mathbf{x}_2 son estadísticamente independientes [21]:

$$\lambda(x_1 \oplus x_2) = \log \frac{e^{\lambda(x_1)} + e^{\lambda(x_2)}}{1 + e^{\lambda(x_1) + \lambda(x_2)}} \quad (\text{B.3})$$

Utilizando las relaciones $\tanh\left(\frac{x}{2}\right) = \frac{e^x - 1}{e^x + 1}$ y $\log \frac{1+x}{1-x} = 2 \tanh^{-1} x$:

$$\begin{aligned} \log \frac{e^{\lambda(x_1)} + e^{\lambda(x_2)}}{1 + e^{\lambda(x_1) + \lambda(x_2)}} &= \log \frac{(e^{\lambda(x_1)} + 1)(e^{\lambda(x_2)} + 1) - (e^{\lambda(x_1)} - 1)(e^{\lambda(x_1)} - 1)}{(e^{\lambda(x_1)} + 1)(e^{\lambda(x_2)} + 1) + (e^{\lambda(x_1)} - 1)(e^{\lambda(x_1)} - 1)} = \\ &= -\log \frac{1 + \tanh(-\lambda(x_1)/2) \tanh(-\lambda(x_2)/2)}{1 - \tanh(-\lambda(x_1)/2) \tanh(-\lambda(x_2)/2)} = \\ &= -2 \tanh^{-1}(\tanh(-\lambda(x_1)/2) \tanh(-\lambda(x_2)/2)) \end{aligned} \quad (\text{B.4})$$

En el caso de n variables estadísticamente independientes:

$$\lambda \left(\sum_{j=1}^n \oplus x_n \right) = -2 \tanh^{-1} \left(\prod_{j=1}^n \tanh(-\lambda(x_n)/2) \right) \quad (\text{B.5})$$

Apéndice C

Tiempos de decodificación

C.1 SC-LDPC(3,24), L=48

20 iteraciones			40 iteraciones		
Factor de ensanchado M	Tamaño de ventana W	Tiempo de decodificación (s)	Factor de ensanchado M	Tamaño de ventana W	Tiempo de decodificación (s)
640	10	16.50	640	10	32.55
	15	21.80		15	44.44
	20	27.3		20	55.88
	30	40.05		30	82.31
960	10	25.61	960	10	51.05
	15	39.09		15	78.32
	20	60.34		20	124.39
	30	81.86		30	165.33
1280	10	38.01	1280	10	76.69
	15	74.05		15	146.91
	20	103.80		20	209.38
	30	130.90		30	257.59

Tabla C.1: Tiempos de decodificación de palabra código SC-LDPC(3,24) L=48, 20 y 40 iteraciones

C.2 SC-LDPC(4,32), L=47

20 iteraciones			40 iteraciones		
Factor de ensanchado M	Tamaño de ventana W	Tiempo de decodificación (s)	Factor de ensanchado M	Tamaño de ventana W	Tiempo de decodificación (s)
640	10	28.56	640	10	55.98
	15	39.91		15	83.08
	20	54.50		20	109.08
	30	83.61		30	163.58

20 iteraciones			40 iteraciones		
Factor de ensanchado M	Tamaño de ventana W	Tiempo de decodificación (s)	Factor de ensanchado M	Tamaño de ventana W	Tiempo de decodificación (s)
960	10	45.61	960	10	93.83
	15	83.36		15	171.23
	20	122.66		20	252.39
	30	151.98		30	307.97
1280	10	78.46	1280	10	158.82
	15	147.12		15	297.15
	20	191.96		20	384.03
	30	240.62		30	482.57

Tabla C.2: Tiempos de decodificación de palabra código SC-LDPC(4,32) L=47, 20 y 40 iteraciones

C.3 SC-LDPC(3,15), L=48

20 iteraciones			40 iteraciones		
Factor de ensanchado M	Tamaño de ventana W	Tiempo de decodificación (s)	Factor de ensanchado M	Tamaño de ventana W	Tiempo de decodificación (s)
640	10	8.99	640	10	16.11
	15	12.87		15	20.70
	20	15.67		20	24.70
	30	25.79		30	41.56
960	10	14.30	960	10	26.98
	15	23.31		15	39.19
	20	38.88		20	65.42
	30	46.52		30	72.52
1280	10	22.80	1280	10	41.22
	15	46.84		15	79.60
	20	59.31		20	105.24
	30	65.22		30	108.45

Tabla C.3: Tiempos de decodificación de palabra código SC-LDPC(3,15) L=48, 20 y 40 iteraciones

C.4 SC-LDPC(4,20), L=47

20 iteraciones			40 iteraciones		
Factor de ensanchado M	Tamaño de ventana W	Tiempo de decodificación (s)	Factor de ensanchado M	Tamaño de ventana W	Tiempo de decodificación (s)
640	10	14.19	640	10	25.41
	15	20.19		15	33.97
	20	28.65		20	48.51
	30	42.78		30	69.68
960	10	23.15	960	10	42.59
	15	44.81		15	78.53
	20	64.04		20	108.78
	30	74.67		30	118.13
1280	10	41.60	1280	10	73.18
	15	79.17		15	131.01
	20	100.60		20	169.59
	30	118.18		30	191.18

Tabla C.4: Tiempos de decodificación de palabra código SC-LDPC(4,20) L=47, 20 y 40 iteraciones

BIBLIOGRAFÍA

- [1] A. Artés, F. Pérez, J. Cid, R. López, C. Mosquera, F. Pérez. “Comunicaciones Digitales”. Capítulo 10: “Codificación para protección contra errores”, Pearson Educación, 2007
- [2] D.J Costello, L. Dolecek, T. E. Fuja, J. Klierer, D. G. M. Mitchell, R. Smarandache. “Spatially Coupled Sparse Codes on Graphs: Theory and Practice”, IEEE Communications Magazine, vol. 52, no. 7, July 2014, pp. 168-176
- [3] T. Richardson, R. Urbanke. “Modern Coding Theory”, Cambridge University Press, 2008
- [4] Recomendación UIT-T G.709/Y.1331 “Interfaces para la Red Óptica de Transporte”, 2012
- [5] Recomendación UIT-T G.975 “Corrección de errores en recepción para sistemas submarinos”, 2001
- [6] Recomendación UIT-T G.975.1 “Corrección de errores en recepción para sistemas submarinos con multiplexación por división en longitud de onda densa de alta velocidad binaria”, 2004
- [7] Recomendación UIT-T G.707/Y.1322 “Interfaz de nodo de red para la jerarquía digital síncrona”, 2007
- [8] T. Mizouchi. “Soft-decision FEC for 100 Gb/s DSP based Transmission”, IEEE Photonics Society Summer Topical Meeting, July 2009, pp. 107-108
- [9] W. D. Grover. “Forward Error Correction in Dispersion-limited Lightwave Systems”, Journal of Lightwave Technology, vol. 6, Issue 5, May 1988, pp. 643-654
- [10] F. Chang, K. Onohara, T. Mizouchi. “Forward Error Correction for 100G Transport Networks”, IEEE Communications Magazine, vol. 48, Issue 3, Mar. 2010, pp 48-55
- [11] A. Puc, F. Kerfoot, A. Simons, D. L. Wilson. “Concatenated FEC Experiment over 5000 km Long Straight Line WDM Test Bed”, Optical Fiber Communication Conference, Feb. 1999, pp. 255-258

- [12] T. Mizouchi, K. Ouchi, T. Kohayashi, Y. Miyata, K. Kuno, H. Tagami, K. Kuho, H. Yoshida, M. Akita, K. Motoshima. "Experimental Demonstration of Net Coding Gain of 10.1 dB using 12.4 Gb/s Block Turbo Codes with 3-bit Soft-Decision", Optical Fiber Communications Conference, OFC 2003, Mar. 2003, pp. PD21-P1-3
- [13] K. Ouchi, K. Kubo, T. Mizuochi, Y. Miyata, H. Yoshida, H. Tagami, K. Shimizu, T. Kobayashi, K. Shimomura, K. Onohara, K. Motoshima. "A Fully Integrated Block Turbo Code FEC for 100 Gb/s Optical Communication Systems", Optical Fiber Communication Conference, Mar. 2006
- [14] Optical Internetworking Forum (OIF). "100G Forward Error Correction white paper" [Online]. Disponible: <http://www.oiforum.com>
- [15] D. Chang, F. Yu, Z. Xiao, Y. Li, N. Stojanovic, C. Xie., X. Shi, X. Xu, Q. Xiong. "FPGA Verification of a single QC-LDPC Code for 100 Gb/s Optical Systems without Error Floor down to BER of 10^{-15} ", Optical Fiber Communication Conference and Exposition, Mar. 2011, pp. 1-3
- [16] S. Dave, L. Esker, F. Mo, W. Thesling, J. Keszenheimer, R. Fuerst. "Soft-decision Forward Error Correction in a 40 nm ASIC for 100-Gbps OTN Applications", Optical Fiber Communication Conference and Exposition, Mar. 2011, pp. 1-3
- [17] Y. Miyata, K. Sugihara, W. Matsumoto, K. Onohara, T. Sugihara, K. Kubo, H. Yoshida, T. Mizuochi. "A Triple-Concatenated FEC using Soft-Decision Decoding for 100 Gb/s Optical Transmission", Optical Fiber Communication Conference, Mar. 2011
- [18] I. B. Djordjevic. "Advances in Error Correction Coding for High-Speed Optical Transmission", IEEE Photonics Conference (IPC 2013), Sept. 2013, pp. 133-134
- [19] S. Che, X. Wang. "Masking Matrix Design of Quasi-Cyclic LDPC Codes for High-speed Optical Communications", International Journal of Future Computer and Communication, vol.2, no.4, Aug. 2013, pp. 296-298.
- [20] B. P. Smith, A. Farhood, A. Hunt, F. R. Kschischang, J. Lodge. "Staircase codes: FEC for 100 Gb/s OTN", Journal of Lightwave Technology, vol. 30, Issue 1, Jan. 2012, pp. 110-117

- [21] T. K. Moon “Error Correction Coding. Mathematical Methods and Algorithms”. Capítulo 15: “Low Density Parity-Check Codes”, Wiley-Interscience, 2005, pp. 634-653
- [22] D. G. M. Mitchell, A. E. Pusane, K. S. Zigangirov, D. J. Costello. “Asymptotically good LDPC convolutional codes based on protographs,” *Information Theory*, Jul. 2008, pp. 1030 – 1034
- [23] M. Lentmaier, G. P. Fettweis, K. S. Zigangirov, D. J. Costello. “Approaching capacity with asymptotically regular LDPC codes,” *Information Theory and Applications Workshop*, Feb. 2009, pp. 173–177
- [24] A. R. Iyengar, M. Papaleo, P. H. Siegel, J. K. Wolf, A. Vanelli-Coralli, G. E. Corazza. “Windowed Decoding of Protograph-based LDPC Convolutional Codes over Erasure Channels”, *IEEE Transactions on Information Theory*, vol. 58, Issue 4, April 2012, pp. 2303-2320
- [25] B. Vasic, I. B. Djordjevic, R. K. Kostuk. “Low-Density Parity Check Codes and Iterative Decoding for Long-Haul Optical Communication Systems”, *Journal of Lightwave Technology*, vol. 21, no. 2, Feb.2003, pp. 438-446